Further properties of random orthogonal matrix simulation

Daniel Ledermann a, Carol Alexander b,∗

a ICMA Centre, Henley Business School at Reading, Reading RG6 6BA, UK
b School of Business, Management and Economics, University of Sussex, Falmer, Brighton, BN1 9SK, UK

Received 18 June 2011; received in revised form 25 July 2012; accepted 28 July 2012
Available online 23 August 2012

Abstract

Random orthogonal matrix (ROM) simulation is a very fast procedure for generating multivariate random samples that always have exactly the same mean, covariance and Mardia multivariate skewness and kurtosis. This paper investigates how the properties of parametric, data-specific and deterministic ROM simulations are influenced by the choice of orthogonal matrix. Specifically, we consider how cyclic and general permutation matrices alter their time-series properties, and how three classes of rotation matrices – upper Hessenberg, Cayley, and exponential – influence both the unconditional moments of the marginal distributions and the behaviour of skewness when samples are concatenated. We also perform an experiment which demonstrates that parametric ROM simulation can be hundreds of times faster than equivalent Monte Carlo simulation.

AMS codes: 15-04; 15B10; 15B52; 65-04; 65F25; 65F30; 65F60

Keywords: Computational efficiency; L matrix; Random orthogonal matrix (ROM); Rotation matrix; Simulation

1. Introduction

Many real-world prediction problems are resolved with reference to a single, observed multivariate sample. For instance, the sample may be used to estimate model parameters as in the ensemble simulation models that are used for climate, flood and biodiversity predictions – see [30,20] and many others since. Or, an observed sample of historical values might simply be used to build a non-parametric, empirical distribution which is then assumed to represent a distribution for future values. This approach is typically applied to resolve financial prediction problems, where it is commonly termed ‘historical simulation’. For instance, despite the limitations discussed in [32], the survey of [31] shows that about three-quarters of large commercial banks estimate their minimum regulatory capital using historical simulation.

Random orthogonal matrix (ROM) simulation is a new procedure, introduced by [22], that can generate any number of multivariate random samples with mean, covariance and Mardia [26] higher moments exactly the same as those of an observed sample. This way, ROM simulation overcomes the standard criticism of historical simulation, i.e. the implicit assumption that only what is historically observed can determine the future. The multivariate moments are often regarded as the salient characteristics of a system, and every ROM simulation will have exactly the same mean,

∗ Corresponding author.
E-mail address: c.o.alexander@sussex.ac.uk (C. Alexander).

0378-4754/$36.00 Crown Copyright © 2012 Published by Elsevier B.V. on behalf of IMACS. All rights reserved.
http://dx.doi.org/10.1016/j.matcom.2012.07.013
covariance matrix and Mardia skewness and kurtosis as those of the historically observed sample. But randomness is introduced to each simulation via the application of a random orthogonal matrix. This type of ROM simulation is called data-specific ROM simulation.

In other practical applications it is common to resolve prediction problems using parametric Monte Carlo simulation. In this context specific values are assigned to the parameters of an assumed form of multivariate distribution, such as the mean vector and covariance matrix of a multivariate normal (MVN) simulation. Parametric ROM simulation is based on a preliminary Monte Carlo simulated sample. Each ROM simulation will have exactly the same sample mean and covariance as the theoretical mean and covariance, and their higher Mardia moments will be identical but equal to those of the preliminary sample. Hence, although ROM simulations have no sampling error in the mean and covariance, the Monte Carlo sampling error in higher moments will be inherited by the ROM simulations.

In fact, ROM simulation does not need to be based on a sample at all, either empirically observed or simulated from a parametric distribution. Deterministic ROM simulation is only based on some pre-specified 'target' multivariate moments, without any parametric or non-parametric distribution. In this case some integer parameters, including the sample size, $m$, of the ROM simulations, are first calibrated to match the target skewness and kurtosis. The ROM simulations again have exactly the target mean and covariance matrix, and they all have the same skewness and kurtosis as the moments based on the calibrated parameter values (typically, a small calibration error arises from the integer parameter constraints).

If the calibrated value of $m$ is small then concatenation may be used to increase the sample size of deterministic ROM simulations.\(^1\) [22] prove that the mean, covariance and kurtosis are invariant under sample concatenation, but the skewness is not. However, once the number of concatenations is fixed, the repeated (concatenated) ROM simulations all have identical skewness.

The aim of this paper is to introduce further statistical, algebraic and computational properties of ROM simulation beyond those introduced in [22]. We investigate how the unconditional moments of the marginal distributions are affected by the choice of rotation matrix which has different random elements in each ROM simulation. We examine the time-series characteristics of ROM simulations, and provide a linear algebraic description of the effect of concatenation on the multivariate skewness, with empirical examples. We also perform a computational experiment which demonstrates that ROM simulation can be hundreds of times faster than equivalent Monte Carlo simulations with exact mean and covariance matrix.

Simulations on multivariate distributions are fundamental to numerous problems in finance. Portfolio returns forecasting methods are commonly based on a parametric model, e.g. from the class of multivariate generalized autoregressive conditional heteroscedasticity (GARCH) models, which capture the volatility and correlation clustering which is usually present in financial risk factor returns, but nevertheless requires time-consuming simulations for forecasting – see [37,10,38] and many others. Even in problems where only the terminal distribution is modelled (as it is when pricing European options, for example) simulation is still required under non-affine continuous-time models: see [33] for advanced simulation methods and see [4,21] for the evaluation of various non-affine models.

Simulation is also the resolution method of choice for many financial institutions employing pricing models with non-standard distributions, see [19] – or for pricing options with exotic or path-dependent pay-offs, see [8,11]. Multivariate simulations are applied in many other areas of finance, for instance to the design of algorithmic trading algorithms as in [35] and to the assessment of portfolio Value-at-Risk (VaR) for which the academic literature is prolific – see [3,6,28] and many others. When VaR models are extended to stress testing, a typical approach is to target values of multivariate moments which reflect increased volatilities, correlations, skewness and kurtosis. Deterministic ROM simulation clearly has a natural application in this context.

Given the wealth of potential applications to finance, our empirical examples on calibration and time-series properties will be based on financial data, and on a simulation from the Markov-switching model, introduced by [14], with two asymmetric GARCH components. In the following: Section 2 provides an overview of the ROM simulation methodology; Section 3 explains the methods we use to generate random rotation matrices; Section 4 investigates the higher multivariate moments of concatenated ROM simulations, which can be used to generate random samples with arbitrarily large size using small-sample data-specific $L$ matrices, or deterministic $L$ matrices; Section 5 examines the

---

\(^1\) Concatenation may also be applied to data-specific ROM simulations if the observed sample size is small, this way increasing the sample size of historical simulations.
time-series properties of ROM simulations; Section 6 explores the effect that different rotation matrices have on the marginal properties of parametric ROM simulation and Section 7 examines the robustness of those results to the choice of correlation matrix; Section 8 compares the computational efficiency of rotational parametric ROM with Monte Carlo simulations and Section 9 summarizes and concludes. An Appendix A illustrates the calibration of deterministic L matrices.

2. Methodology

A special rectangular orthogonal matrix called an L matrix, introduced by [22], is fundamental to ROM simulation. The methodology produces random samples \( X_{mn} \), of size \( m \) on \( n \) linearly independent random variables, each with the same sample mean vector \( \mu_n \) and covariance matrix \( S_n \). If the L matrix is generated by orthogonalizing an empirically observed or Monte Carlo simulated sample, then the Mardia [26] higher moments of each ROM simulation are identical to the corresponding moments of the sample. Alternatively, a deterministic L matrix may be used. Such a matrix has one or more integer parameters that are calibrated to target some pre-assigned values for Mardia skewness and kurtosis. The integer parameter constraints can lead to some error between the skewness and kurtosis of the ROM simulations and the target values, but typically this error is small.

ROM simulation is most easily introduced by considering the MVN case. Monte Carlo simulation produces a sample with mean vector and covariance matrix different from the population (theoretical) parameters, due to sampling error. This is one of the reasons why a very large number of simulations is usually employed. However, it is possible to perform a simple affine transform on each simulation so that it has sample mean vector and sample covariance matrix that exactly match the theoretical parameters, \( \mu_n \) and \( S_n \), as follows. Assuming there is no exact linear relation between the \( n \) random variables in the system (in which case one or more variables could be eliminated) their covariance matrix \( S_n \) will be positive definite. Hence, we can always find a decomposition of the form \( S_n = A_n^T A_n \), where \( A_n \) is a Cholesky matrix, for example.\(^2\) We can therefore apply the following transformation to any Monte Carlo simulation \( X_{mn} \) of size \( m \) on the \( n \) random variables:

\[
L_{mn} = m^{-1/2}(X_{mn} - \text{1}_m \mu_n')A_n^{-1}.
\]

The inverse transform is

\[
X_{mn} = \text{1}_m \mu_n' + m^{1/2}L_{mn}A_n.
\]

It is now straightforward to show that \( X_{mn} \) will have mean \( \mu_n \) and covariance matrix \( S_n \) if, and only if, the matrix \( L_{mn} \) satisfies the following conditions:

\[
L_{mn}' L_{mn} = \text{1}_n \quad \text{with} \quad 1_m L_{mn} = 0_n'.
\]

Any matrix \( L_{mn} \) satisfying (1) is called an L matrix. That is, an L matrix is a special type of rectangular orthogonal matrix – one whose columns sum to zero. Since the columns of \( L_{mn} \) are linearly independent by definition, and the nullity of \( L_{mn}' \) is at least one, the rank-nullity theorem implies that, for any \( L \) matrix, \( m > n \).

\( L \) matrices can be found by orthogonalizing any linearly independent set of contrasts, i.e. subsets \( \mathcal{H} \subseteq \mathbb{R}^m \), defined by:

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

In other words, a contrast is a column vector whose elements sum to zero. Typically, \( L \) matrices are constructed by taking a pair \((m, n) \in \mathbb{Z}_+^2\), with \( n < m \), picking \( N(m) \) linearly independent vectors in \( \mathcal{H} \), where \( n \leq N(m) < m \), and using these vectors to form the columns of a matrix \( V_{m,N(m)} \). Then we apply the Gram–Schmidt procedure to \( V_{m,N(m)} \) to produce a matrix \( W_{m,N(m)} = GS(V_{m,N(m)}) \), with orthonormal columns. Finally, we select any \( n \) columns from \( W_{m,N(m)} \) to form the \( L \) matrix, \( L_{mn} \).

The properties of an \( L \) matrix are inherited from the contrasts that are used in its construction. For instance, data-specific \( L \) matrices \( L_{mn} \) are formed from mean deviations of an empirically observed sample matrix \( X_{mn} \), representing

\(^2\) Another alternative would be to set \( S_n = Q_n A_n Q_n' \), where \( A_n \) is the diagonal matrix of eigenvalues, and \( Q_n \) is the orthogonal matrix of eigenvectors of \( S_n \), so that \( A_n = A_n^\dagger Q_n' \).
it is easy to show that

\[ m \] observations on \( n \) linearly independent random variables. Parametric \( L \) matrices are formed from a sample \( X_{mn} \) on a given multivariate distribution. For instance, suppose \( X_{mn} \sim D(\mu_n, S_n) \), where \( D \) is an elliptical distribution. In each case the columns of \( X_{mn} \) usually do not have zero sample mean, hence define \( V_{mn} = X_{mn} - I_m X_n \), so \( V_{mn} \) is a contrast matrix. Then \( L_{mn} = GS(V_{mn}) \) is an \( L \) matrix.

Deterministic \( L \) matrices are not constructed with reference to a sample. Instead, we directly specify a set of deterministic vectors satisfying (1). For instance, \( L_{mn} = (\ell_1, \ldots, \ell_n) \), where

\[
\ell_j = \left( (m-n+j-1)(m-n+j) \right)^{-1/2} \begin{bmatrix} (m-n+j-1), & 1, & \ldots, & 1, & -(m-n+j-1), & 0, & \ldots, & 0 \end{bmatrix}^T_{n-j} \tag{2}
\]

for \( 1 \leq j \leq n \).

Following [22], Ledermann matrices \( L_{mn} \) consist of any \( n \) columns of the matrix defined by (2). Since \( n \) is fixed to be the number of variables, the number of observations \( m \) becomes a parameter which could be calibrated so that ROM simulations derived from these matrices target some pre-specified Mardia higher moments as closely as possible. Note that, since all \( L \) matrices must satisfy (1), the mean and covariance of every ROM simulation will always match the target mean vector \( \mu_n \) exactly.

Introducing a further, positive integer parameter, \( k \), allows ROM simulations based on deterministic \( L \) matrices to target the Mardia moments more precisely. This parameter may be introduced in different ways, thus defining three other types of deterministic \( L \) matrices, denoted \( L_{mn}^k \). To construct a Type I deterministic \( L \) matrix we set \( N(m) = m+1-2k \) and, to ensure that \( n \leq N(m) \), we require \( 2k \leq m+1-n \). Then set \( V_{mn,N(m)} = (v_1, \ldots, v_{N(m)}) \), where

\[
v_j = \begin{bmatrix} 0, & \ldots, & 0, & 1, & \ldots, & 1, & -1, & \ldots, & -1, & 0, & \ldots, & 0 \end{bmatrix}^T_{2k} \quad \text{for } 1 \leq j \leq N(m). \tag{3}
\]

A Type II matrix is defined by requiring \( N(m) = m-k \geq n \). In this case the GS pre-image matrix \( V_{mn,N(m)} = (v_1, \ldots, v_{N(m)}) \) is defined by

\[
v_j = \begin{bmatrix} 0, & \ldots, & 0, & 1, & \ldots, & 1, & -k, & 0, & \ldots, & 0 \end{bmatrix}^T_{k} \quad \text{for } 1 \leq j \leq N(m). \]

For a Type III matrix we set \( N(m) = m-2 \), require that \( n \leq N(m) \), and define \( V_{mn,N(m)} = (v_1, \ldots, v_{N(m)}) \) where

\[
v_j = \begin{bmatrix} 0, & \ldots, & 0, & k, & -1, & \ldots, & -k, & 0, & \ldots, & 0 \end{bmatrix}^T_{m-2} \quad \text{for } 1 \leq j \leq N(m). \]

In each case the columns of \( V_{mn} \) form a linearly independent set in \( \mathcal{H} \) and the \( L_{mn}^k \) matrix of each type is any \( n \) columns of the GS image of \( V_{mn,m+1-2k} \) (Type I), \( V_{mn,m-k} \) (Type II) or \( V_{mn,m-2} \) (Type III). The numbers of variables \( n \) is fixed but the parameters \( m \) and \( k \) of Type I, II and III deterministic \( L \) matrices can be set to any integer values such that: \( m+1-2k \geq n \) (Type I), \( m-k \geq n \) (Type II) and \( m-2 \geq n \) (Type III), with each of these conditions following from the rank-nullity theorem. In Appendix A we give an empirical example of the calibration of \( m \) and \( k \) for a Type I \( L \) matrix to target some pre-specified values for the Mardia skewness and kurtosis.

Now let \( Q_m \) be an \( m \times m \) permutation matrix and \( R_n \) be any \( n \times n \) orthogonal matrix. Then, given any \( L \) matrix \( L_{mn} \) it is easy to show that \( Q_m L_{mn} R_n \) is also an \( L \) matrix. The fundamental idea of ROM simulation is to use randomly generated elements in the matrices \( Q_m \) and \( R_n \). That is, starting with any (data-specific, parametric or deterministic) \( L \) matrix, the corresponding ROM simulation generates random samples \( X_{mn} \) via the equation:

\[
X_{mn} = L_{mn} \mu_n + \sqrt{m} Q_m L_{mn} R_n A_n, \tag{4}
\]

where \( Q_m \) is a random permutation matrix, \( R_n \) is any random orthogonal matrix and \( A_n A_n^T = S_n \).

The ROM simulation properties of different classes of \( L \) matrices were discussed in [22]. This paper focuses on the ROM simulation properties of the random orthogonal matrices, \( Q_m \) and \( R_n \). We consider cyclic and general permutation matrices for \( Q_m \) and three different types of rotation matrices for \( R_n \). Note that permutation matrices are only applied to permute the rows of the \( L \) matrix (i.e. to change the ordering of observations in a time-series sample) because there is

---

3 We remark that the matrix (2) is the last \( m-1 \) columns of a type of \( m \times m \) generalised Helmert matrix, employed in statistical contrast analysis. The Type I \( L \) matrix (3) with \( k = 1 \) has also applied in the same context. It is called the ‘profile contrast matrix’ in SAS.
usually no point to randomly permute the columns (i.e. to change order of the variables). Hence permutation matrices only affect the time-series properties of ROM simulations, whereas the random rotation matrices primarily influence the unconditional moments of the marginal distributions.

3. Generating random rotation matrices

A rotation matrix \( R_n \) is a square orthogonal matrix with determinant one. In this paper we aim to link the form of rotation matrix to the characteristics of ROM simulations and we choose to focus on the three different classes of rotations that are well established in the literature: i.e. (i) upper Hessenberg, (ii) Cayley and (iii) exponential. The effect of other rotation matrices on the characteristics of ROM simulation is left to further research. For instance, a random rotation matrix may be generated by orthogonalizing a square matrix of independent random numbers and, if its determinant is \(-1\), multiplying the elements by \(-1\). Depending on the distribution of the random numbers a great variety of rotation matrices can be generated this way.

An upper Hessenberg matrix \( H_n \) of degree \( n \), is a matrix with zeros below the first sub-diagonal. [5, 12] and many other authors show that we can construct orthogonal upper Hessenberg matrices using a particular type of [13] rotation, \( G_n(\theta_j) \) which is equal to the \( n \times n \) identity matrix everywhere, apart from the \( 2 \times 2 \) principal submatrix which satisfies:

\[
G_n(\theta_j)[j, j + 1; j, j + 1] = \begin{pmatrix}
\cos(\theta_j) & \sin(\theta_j) \\
-\sin(\theta_j) & \cos(\theta_j)
\end{pmatrix}.
\]

We use random orthogonal upper Hessenberg matrices \( H_n \) which are formed by taking the product of \( n - 1 \) Givens rotations: \( H_n = G_n(\theta_1)G_n(\theta_2)\ldots G_n(\theta_{n-1}) \), where \( \theta_j \) is chosen randomly in the interval \([0, 2\pi]\) for \( 1 \leq j \leq n - 1 \). In our \( n \)-dimensional ROM simulations (4) we will apply orthogonal matrices \( R_n \) which are products of \( n - 1 \) upper Hessenberg matrices. In general these have no zero elements, since the product of \( k \) upper Hessenberg matrices only has zeros below the \( k \)th sub-diagonal.

Cayley rotations are derived from skew-symmetric matrices (i.e square matrices \( A_n \) having the property \( A_n' = -A_n \)) via the [9] transform \( R_n = (I_n - A_n)^{-1}(I_n + A_n) \). This is well defined since \( (I_n - A_n) \) has non-zero determinant for all skew-symmetric matrices \( A_n \), and it is also invertible unless \( R_n \) has an eigenvalue of \(-1\). Clearly \( R_nR_n' = I_n \), because \( (I_n - A_n) \) and \( (I_n + A_n) \) commute, so \( R_n \) is an orthogonal matrix; and indeed it is also a rotation, since \( \det(R_n) = \det(I_n + A_n)/\det(I_n - A_n) = 1 \). Generating random, skew-symmetric matrices is straightforward because skew-symmetric matrices of order \( n \) correspond to vectors of length \( n(n - 1)/2 \) in the following way: let \( a_{N(n)} \) be a random vector containing \( N(n) = n(n - 1)/2 \) elements. We start by labelling these entries using indices \((i, j)\) satisfying \( 1 \leq i < j \leq n \). That is, \( a_{N(n)} = \left( a_{12}, \ldots, a_{1n}, a_{23}, \ldots, a_{2n}, \ldots \right) \). This determines a unique skew-symmetric matrix \( A(a_{N(n)}) \) whose elements are defined as

\[
[A(a_{N(n)})]_{ij} = \begin{cases} 
a_{ij} & i < j, \\
0 & i = j, \\
-a_{ij} & i > j,
\end{cases} \quad \text{for } 1 \leq i, j \leq n. \tag{5}
\]

It is also possible to transform skew-symmetric matrices \( A_n \) into orthogonal matrices using the matrix exponential function:

\[
\exp(A_n) = \sum_{k=0}^{\infty} \frac{A_n^k}{k!}.
\]
When $A_n$ is skew-symmetric $\exp(A_n)$ is orthogonal, since $\exp(A_n) \exp(A_n)' = \exp(A_n + A_n') = \exp(0_n) = I_n$, and it is a rotation since $\det(\exp(A_n)) = \exp(\text{tr}(A_n)) = e^0 = 1$. Calculating matrix exponentials presents computational challenges (see [29]). A popular approach utilises the spectral decomposition of a matrix and the property that if $B_n$ is invertible then $\exp(A_n)B_n^{-1} = \exp(B_nA_nB_n^{-1})$. It assumes that $A_n$ has $n$ linearly independent eigenvectors, and can be written in the form $A_n = P_nA_nP_n^{-1}$, where $P_n$ is the matrix of eigenvectors and $A_n = \text{diag}(\lambda_1, \ldots, \lambda_n)$ is the corresponding diagonal matrix of eigenvalues. Then the exponential of $A_n$ can be calculated as:

$$\exp(A_n) = P_n^{-1} \exp(P_nA_nP_n^{-1})P_n = P_n^{-1} \exp(A_n)P_n = P_n^{-1} D_n P_n,$$

(6)

where $D_n = \text{diag}(\exp(\lambda_1), \ldots, \exp(\lambda_n))$. A potential problem with method (6) is that it relies on $A_n$ having $n$ linearly independent eigenvectors, to ensure that $P_n$ is invertible. If $P_n$ is close to being singular the calculation of $P_n^{-1}$ may be numerically unstable. However, $A_n$ is skew-symmetric and so has a complete set of unitary eigenvectors. Thus this approach is perfectly stable and may well be the fastest.\(^6\)

An alternative method for computing matrix exponentials exploits the fact that $\exp(A_n) = (A_n/k)^k$. It is common to choose the scalar $k = 2^j$ to be the smallest power of two such that the norm of $A_n/k$ is less than or equal to one. Under these conditions $\exp(A_n/k)$ can be computed more reliably using either Taylor or Padé approximants, see [29]. Squaring the resulting matrix $j$ times produces a good approximation for the desired matrix exponential $\exp(A_n)$. This procedure is implemented in the MATLAB matrix exponential function. For a recent survey of matrix exponential and other numerical methods for computing matrix functions see [18].\(^7\)

4. Sample concatenation

In some contexts it is advantageous to concatenate ROM simulations, i.e. to put several simulations together to form one simulation with many more observations. For instance, an historical sample that is used to generate a data-specific matrix may have fewer observations than desired, as historical data are sometimes difficult to obtain. Then small-sample bias would arise, in addition to the ubiquitous single-sample problems. The other case when one might wish to concatenate simulations is when the parameter $m$ that is calibrated for a deterministic $L$ matrix is relatively small. For example, the first calibration illustrated in Appendix A yields the value $m = 183$, but simulations with only 183 observations are rarely used in financial applications.

It is proved in [22], Proposition 2.2, that ROM simulations may repeated many times and concatenated without affecting the mean and covariance matrix of the concatenated sample. Moreover, by the same proposition we know that the Mardia kurtosis will also remain invariant provided the simulations are all based on the same $L$ matrix. That is, ignoring the random permutation matrix without loss of generality, our concatenated ROM simulated sample takes the form

$$X_{rm,n} = 1_{rm} R_n' + \sqrt{m} \left( (L_{mn} R_n^{(1)})', \ldots, (L_{mn} R_n^{(r)})' \right)' A_n.$$

(7)

Using the affine invariance property of multivariate kurtosis, each sub-sample in (7) will have identical kurtosis, even if the random orthogonal matrices $R_n^{(1)}, \ldots, R_n^{(r)}$ are of different types.

However, even when the repeated ROM simulations are all based on the same $L$ matrix with skewness $\tau$, and the random rotation matrices in (7) are all of the same type, the concatenated sample $X_{rm,n}$ will have a different Mardia skewness. To investigate this skewness we suppose, without loss of generality, that all our samples have zero mean vector and covariance matrix equal to the identity.\(^8\) Then, using (7), we can write this concatenated sample in the simplified form

$$X_{rm,n} = ((X_{mn}^{(i)})', \ldots, (X_{mn}^{(r)})')', \text{ where } X_{mn}^{(i)} = \sqrt{m} L_{mn} R_n^{(i)} \text{ for } i = 1 \ldots r.$$

We now want to relate the skewness of $X_{rm,n}$ to the skewness of the underlying $L$ matrix $L_{mn}$. Of course, since the $X_{mn}^{(i)}$ are all non-singular affine transformations of $L_{mn}$, they all have the same multivariate skewness. In particular,

\(^6\) Many thanks to Prof. Nick Higham for pointing this out.

\(^7\) The MATLAB function “expm” is based on the algorithm in [17]. An improvement is given by [1]. Code is available from http://www.maths.manchester.ac.uk/higham/papers/matrix-functions.php.

\(^8\) That this is without loss of generality follows from the affine invariance property of Mardia skewness measures.
\( \tau(X_{imn}) = \tau(L_{mn}) \) for \( 1 \leq i \leq r \). To investigate the behaviour of skewness under concatenation we first apply [22], Proposition 2.2, to write

\[
\tau(X_{imn}) = r^{-1} \tau(L_{mn}) + 8r^{-2} \sum_{k<l} \tau_C(L_{mn}R_{kn}^{(k)}, L_{mn}R_{ln}^{(l)}),
\]

(8)

where \( \tau_C \) is the co-skewness defined in [22], Definition 2.1. The summation on the right hand side of Eq. (8) contains \( r(r-1)/2 \) terms. We therefore define

\[
\overline{C} = \frac{2}{r(r-1)} \sum_{k<l} \tau_C(L_{mn}R_{kn}^{(k)}, L_{mn}R_{ln}^{(l)}),
\]

and interpret this quantity as the average co-skewness between samples of the form \( L_{mn}R_{kn}^{(k)} \) and \( L_{mn}R_{ln}^{(l)} \). Substituting this average into (8) we find that

\[
\tau(X_{imn}) = r^{-1} \tau(L_{mn}) + 4r^{-1}(r-1)\overline{C}.
\]

(9)

On re-arranging (9), it is clear that:

\[
\tau(X_{imn}) \text{ will be strictly less than } \tau(L_{mn}) \text{ for all } r > 0 \text{ if and only if } \overline{C} < \tau(L_{mn})/4,
\]

where \( \tau(L_{mn})/4 \) is easily shown to be the co-skewness between identical samples. First we investigate this inequality empirically, by generating sample pairs \( L_{mn}R_{kn}^{(k)} \) and \( L_{mn}R_{ln}^{(l)} \) from which we calculate \( \overline{C} \). We do this by taking the average of one hundred co-skewness values of the form \( \tau_C(L_{mn}R_{kn}^{(k)}, L_{mn}R_{ln}^{(l)}) \), where \( L_{mn} \) is a Ledermann matrix.\(^9\)

Each term is generated by simulating random orthogonal matrix pairs \( (R_{kn}^{(k)}, R_{ln}^{(l)}) \). Although the matrices \( R_{kn}^{(k)} \) and \( R_{ln}^{(l)} \) are random, we assume that they have the same form because it may be that the type of random rotation matrix used affects the value of \( \overline{C} \). To see whether this is the case we generate three different sets of results, one for each of the rotation matrix types of Section 3.

First note that (9) implies that as \( r \) increases the skewness will level off at a value roughly equal to \( 4\overline{C} \). Fig. 1 depicts how the skewness decreases as the number \( r \) of repetitions increases, for each type of rotation matrix. From this it appears that skewness definitely decreases with \( r \), and that \( \overline{C} \) is greatest for exponential ROM simulations and lowest for upper Hessenberg ROM simulations.

Fig. 1. The relationship between Mardia skewness of ROM simulations and the number of repetitions in the concatenated samples. All concatenations are constructed by multiplying the same Ledermann matrix (\( n = 10 \) and \( m = 50 \)) with random orthogonal matrices of the appropriate form.

\(^{9}\) Then, by Proposition 2.1 of [22] we know that the proposed upper bound \( \tau(L_{mn})/4 \) is \( n[(m-2) + (m-n)-1]/4 \).
Next, for each type of rotation matrix we compute \( \mathcal{C} \) one hundred times. In Table 1 we report the average value of the co-skewness averages \( \mathcal{C} \), along with standard error estimates. From the three cases investigated, average co-skewness is not only greatest but also least variable when samples are constructed using exponential rotation matrices.

Given that \( \mathcal{C} \) is variable, we seek an upper bound for it’s value and then relate this upper bound to the skewness of a concatenated sample using (9). Since \( \mathcal{C} \) is an average of co-skewness terms, we will seek an upper bound for terms of the form \( \tau_C (L_{mn} R_n^{(k)}, L_{mn} R_n^{(l)}) \), where \( R_n^{(k)} \) and \( R_n^{(l)} \) are orthogonal matrices. In Fig. 2, one hundred of such terms are displayed for each of the three rotation matrix forms under investigation. This plot, and all other similarly-generated plots not shown here, reveal that all co-skewness terms \( \tau_C (L_{mn} R_n^{(k)}, L_{mn} R_n^{(l)}) \) appear empirically to be bounded-above by the value \( \tau (L_{mn})/4 \). Thus, the skewness will indeed decrease as the number of repetitions in the concatenated sample increases.

<table>
<thead>
<tr>
<th></th>
<th>Upper Hessenberg</th>
<th>Cayley</th>
<th>Exponential</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu(\mathcal{C}) )</td>
<td>0.02</td>
<td>13.41</td>
<td>26.27</td>
</tr>
<tr>
<td>( s.e(\mathcal{C}) )</td>
<td>2.07</td>
<td>1.88</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Fig. 2. Co-skewness of 100 concatenated ROM simulations and the proposed upper bound \( \tau (L_{mn})/4 \). Each co-skewness term involves the same \( L \) matrix with \( n = 10 \) and \( m = 50 \).

A formal proof of the existence of \( \tau (L_{mn})/4 \) an upper bound reduces to a problem in linear algebra. Since co-skewness is invariant under joint, non-singular affine transformations we can always write

\[
\tau_C (L_{mn} R_n^{(k)}, L_{mn} R_n^{(l)}) = \tau_C (L_{mn}, L_{mn} R_n) \quad \text{where} \quad R_n = (R_n^{(l)}) (R_n^{(k)})'.
\]

Hence it is required to prove that \( \tau_C (L_{mn}, L_{mn} R_n) \leq \tau (L_{mn})/4 \) for any orthogonal matrix \( R_n \). Since both \( L_{mn} \) and \( L_{mn} R_n \) have zero sample means and the identity as sample covariance matrix, we can write

\[
4 \tau_C (L_{mn}, L_{mn} R_n) = m L_m ((L_{mn} R_n' L_{nm}) \circ (L_{mn} R_n' L_{nm}) \circ (L_{mn} R_n' L_{nm})) L_m',
\]

\[
\tau (L_{mn}) = m L_m ((L_{mn} L_{nm}') \circ (L_{mn} L_{nm}') \circ (L_{mn} L_{nm}')) L_m'.
\]

Here, \( \circ \) denotes the Hadamard, or element-wise, matrix multiplication and \( L_m \) is a row vector of ones. Now writing \( A_m = L_{mn} L_{nm}' \) and \( B_m = L_{mn} R_n' L_{nm}' \) it is required to prove that

\[
L_m (A_m \circ A_m) L_m' \geq L_m (B_m \circ B_m \circ B_m) L_m'. \tag{10}
\]
If we can show that \( A_m \circ A_m \circ A_m - B_m \circ B_m \circ B_m \) is positive semi-definite then (10) will follow immediately. The fact that \( A_m - B_m \) is positive semi-definite is an immediate consequence of the Cauchy–Schwartz inequality. However, the proof that \( A_m \circ A_m \circ A_m - B_m \circ B_m \circ B_m \) is also positive definite remains elusive, although it is supported by numerical tests.

If this conjecture holds then \( \tau_C(L_{mn}R(l)_{mn}, L_{mn}R(k)_{mn}) \leq \tau(L_{mn})/4 \) for all orthogonal matrices \( R(l)_{mn} \) and \( R(k)_{mn} \). Then also the act of sample concatenation will always decrease multivariate skewness of the ROM simulations. Indeed skewness will approximately decrease as a function of \( 1/r \) from the initial value \( \tau(L_{mn}) \), where \( r \) denotes the number of samples within the concatenation. Once this number is fixed, repeated concatenated ROM simulations will all have the same skewness, as well as identical means, covariances and kurtosis.

5. Time-series characteristics of ROM simulations

When a sample (column) is endowed with dynamic order it may be regarded as a time series. Time series of returns on a financial asset typically exhibit volatile periods of large changes interspersed with relatively tranquil periods of small changes. This phenomenon, first observed by [25], is commonly referred to as conditional heteroscedasticity or (less formally) as volatility clustering. Clusters in volatility are a sign that the market is experiencing a stressful period, where large negative returns can be followed by further large negative returns, or large positive returns. Thus, the existence of volatility clusters is accompanied by a high autocorrelation in squared returns.

Deterministic \( L \) matrices have a natural structure, with a volatility cluster towards the end of each column. This cluster may be moved to a different, random point in time by the application of a random cyclic permutation matrix; or it may be dispersed throughout the sample by using a random general permutation matrix in the ROM simulation. Further details about time series generation via deterministic ROM simulation are given in [22]. In this section we focus instead on the time-series characteristics of data-specific and parametric ROM simulation.

Post-multiplication of the \( L \) matrix by random rotation matrices \( R_m \) will not affect the autocorrelation function (ACF) of returns, or the ACF of the squared returns. They do affect the empirical ACF, but only by changing the sampling error in the estimation of the autocorrelation coefficients. To demonstrate this in practice, Fig. 3 depicts the ACF of the daily returns (above) and the ACF of the squared daily returns (below) for the UK index, when the multivariate system
of MSCI daily returns is used in data-specific ROM simulation. The different lines correspond to the ACF of the original sample (black), and the ACF of three ROM simulated samples, each based on the data-specific \( L \) matrix but with a different random orthogonal matrix \( R_m \) in (4): upper Hessenberg (blue), Cayley (green) and exponential (red).

Hence, all dynamic characteristics of any type of ROM simulation are related to the pre-multiplication of the \( L \) matrix by random permutation matrices \( Q_m \) in (4). The pre-multiplication of the \( L \) matrix by a random cyclic permutation matrix offset a path’s observations by means of a random shift. Therefore, the dynamic ordering of the observations is maintained. This means that even the empirical ACFs, of both the returns and squared returns, remain unchanged by the application of this type of ROM simulation. By contrast non-cyclic permutation matrices with random elements change the ordering of the observations in a random fashion. Hence, whatever the original ACFs for returns and squared returns, after the application of a general random permutation matrix in ROM simulation both theoretical ACFs will be zero at all lags greater than one.

To illustrate how the time-series properties of ROM simulations are controlled using random permutation matrices \( Q_m \) in (4), we analyse the time-series characteristics of data-specific ROM simulations applied to a long time series of real financial data, viz. twenty years of observed daily returns on the S&P500 index from May 1992 to May 2012. The original time series is depicted in the top graph of Fig. 4. Note that a particularly large volatility cluster is evident during the last quarter of 2008 and the first quarter of 2009, which is the period covering the banking crisis. Then almost exactly three years later another, shorter volatility cluster appears at the onset of the sovereign debt crisis in 2011. Below this we show the time series generated by the data-specific ROM simulation where \( Q_m \) is a random cyclic permutation. Note that the position of all the volatility clusters has been shifted so that the ‘banking crisis’ now occurs during 2002 and the sovereign debt crisis again almost exactly three years afterwards.

A large class of financial applications are suitable for data-specific ROM simulation with random cyclic permutation matrices, namely the specification of dynamic models. A long time-series is typically divided into two periods: the first ‘in-sample’ (or estimation) period is used to estimate the model parameters, and the second ‘out-of-sample’ (or testing) period is used to evaluate the model performance. In many studies of financial model specification several different dynamic models are proposed and a single time series is used to decide which model is best. However, given there is only one time-series of observed data, the performance of each model could be attributed to good or bad luck during the testing period. This observation, made by [36] more than a decade ago, has led to a large literature on ‘data snooping bias’; see [15,34] for example.

![Fig. 4. Daily returns on the S&P500 index (above); a data-specific ROM simulation with \( Q_m \) in (4) being a random cyclic permutation matrix (middle); and a data-specific ROM simulation with \( Q_m \) being a general random permutation matrix (lower).](image-url)

---

10 More details about these data are given in Appendix A.
In this context, an advantage of data-specific ROM simulation with random cyclic permutation matrices is that one can repeatedly re-estimate and re-evaluate the models over different ROM simulations, where in-sample and out-of-sample periods are randomly changed. Further enhancement of data-snooping bias tests could include augmenting the cyclic ROM simulations with random rotation matrices in (4), thus introducing additional randomness to the observations. The role of ROM simulation in reducing data snooping bias is beyond the scope of this paper, but seems a fruitful subject for further research.

The bottom graph in Fig. 4 illustrates a ROM simulation of the S&P 500 data based on a general random permutation matrix. In this case the original dynamic ordering of the returns is changed at random. There will be no autocorrelation in returns, or their squares, except by chance. This type of ROM simulation would be useful in any applications, such as ordinary least squares regression, that require the autocorrelation and conditional heteroscedastic properties of the data to be removed whilst leaving the unconditional empirical distribution of returns unchanged.

Next we analyse the dynamic properties of parametric ROM simulations by generating a preliminary sample, displayed at the top of Fig. 5, consisting of a time series of 2000 observations on the Markov-switching model of [14], with two asymmetric GARCH volatility components. The data generation model for returns at time $t$, $r_t$, is specified, conditional on the information set $I_t = \{r_{t-1}, r_{t-1}, \ldots, r_1\}$, as $r_t|I_t \sim N(0, \sigma^2_t)$ with conditional variance $\sigma^2_t$ switching between two asymmetric GARCH components of the form:

$$\sigma^2_{it} = \omega_i + \alpha_i (r_t - \lambda_i)^2 + \beta_i \sigma^2_{i,t-1}, \quad i = 1, 2.$$  \hspace{1cm} (11)

Model parameters include the transition probability $\pi_{ij}$ of switching to component $j$ when the current state is the GARCH component $i$, for $i = 1, 2$. For the purpose of illustration we have specified the parameters as follows (Table 2):

<table>
<thead>
<tr>
<th>State 1</th>
<th>State 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\omega$</td>
<td>$10^{-8}$</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.05</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.85</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>0.01</td>
</tr>
</tbody>
</table>

This implies a long-term volatility of 11.19% for the low volatility state 1 and 55.23% for the high volatility state 2.\footnote{More details on how these volatilities are calculated can be found in standard textbooks such as [2].} The unconditional state probabilities are 0.75 for state 1 and 0.25 for state 2, with the matrix $(\pi_{ij})$ set to:

$$\pi = \begin{pmatrix} 0.99 & 0.01 \\ 0.03 & 0.97 \end{pmatrix}.$$  

The parameters have been chosen so that simulations from this model have returns with similar characteristics to daily returns on equity indices, such as those shown in Fig. 4. The top graph in Fig. 5 depicts a preliminary simulation from the GARCH model, with 2000 observations. The parametric ROM simulation now proceeds as for the data-specific example given above. The middle graph illustrates the application of a random cyclic permutation matrix. In this case the returns have been randomly shifted forward in time. For instance, the large cluster around observation 500 in the top graph now occurs around observation 750, and all the other returns are shifted forward the same number of time periods so that the dynamic properties of the original sample are left in tact. And the lower graph depicts the application of a general random permutation matrix in the ROM simulation, which destroys all the dynamic structure in the original sample.

As noted in the introduction, GARCH modelling is a popular methodology for resolving numerous financial problems. One can generate many GARCH simulations from the same model, each having volatility clusters that not only occur at different random times, but also have different characteristics. From this perspective the use of GARCH modelling has an advantage over cyclic ROM simulation. However, the downside of using GARCH models is that the assumptions underpinning the model may not hold in practice, and the parameter estimates are subject to sampling...
error. We remark that GARCH model estimation is included in the class of dynamic model specification mentioned above, where cyclic ROM simulations may have an important role to play in the reduction of data snooping bias.

6. Marginal characteristics of ROM simulations

Here we study the effect of rotational matrices on the marginal densities of ROM simulated samples. A great variety of sample characteristics can be generated by choosing a suitable $L$ matrix to use in combination with the random rotational matrix, which we here assume to have either upper Hessenberg, Cayley or exponential form. Our results are organized by considering parametric, deterministic and hybrid $L$ matrices separately, providing illustrations of sample densities, and then summarizing the marginal characteristics at the end of this section. Since we are only concerned with the characteristics of the marginal densities we employ the following test correlation matrix, which was generated randomly in MATLAB, throughout this section.

Table 3
$10 \times 10$ random correlation matrix used for all simulations.

<table>
<thead>
<tr>
<th></th>
<th>1.000</th>
<th>-0.500</th>
<th>0.187</th>
<th>0.238</th>
<th>0.193</th>
<th>0.030</th>
<th>0.394</th>
<th>0.490</th>
<th>-0.348</th>
<th>0.215</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td>0.079</td>
<td>0.180</td>
<td>-0.164</td>
<td>-0.022</td>
<td>-0.287</td>
<td>-0.081</td>
<td>0.127</td>
<td>-0.049</td>
<td>-0.448</td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>-0.195</td>
<td>-0.159</td>
<td>0.417</td>
<td>0.061</td>
<td>0.124</td>
<td>-0.491</td>
<td></td>
<td>-0.048</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>0.018</td>
<td>-0.212</td>
<td>0.085</td>
<td>-0.143</td>
<td>0.278</td>
<td>0.185</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>-0.267</td>
<td>0.420</td>
<td>0.266</td>
<td>-0.048</td>
<td>0.069</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>-0.341</td>
<td>-0.063</td>
<td>-0.587</td>
<td>-0.301</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>-0.138</td>
<td>0.036</td>
<td>0.117</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>-0.398</td>
<td>0.260</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>0.020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.000</td>
</tr>
</tbody>
</table>

6.1. Parametric ROM simulation

Parametric ROM simulation essentially combines a simulated, orthogonalized sample from a given multivariate distribution with random orthogonal matrices. Typically, the single simulated sample is generated using Monte Carlo, orthogonalized to form a parametric $L$ matrix, and then to generate another sample it is multiplied by any random orthogonal matrix. To generate a new sample using traditional Monte Carlo techniques one would have to re-sample from the chosen parametric distribution. In many instances this conventional re-sampling approach is considerably slower than parametric ROM simulation, as we shall see in Section 8.
In this sub-section we conduct experiments on multivariate normal and Student-\(t\) parametric ROM simulations to investigate the marginal distributions of \(X_{mn}\). Here and in the following we focus on three cases, according as the random rotation matrices are (i) of upper Hessenberg form, (ii) a Cayley transformed rotation and (iii) generated using the matrix exponential mapping. Our purpose is to investigate any systematic differences between these three cases, in terms of the characteristics of the ROM-simulated marginals.

In the absence of simulation error in the initial Monte Carlo simulation that generates \(L_{mn}^P\), the marginals of \(L_{mn}^P\) will be normally distributed, and then \(X_{mn}\) defined by (12) will also be multivariate normally distributed. To see this, note that it is clear from Eq. (12) that the columns of our ROM simulated sample \(X_{mn}\) are linear combinations of our (assumed to be normal) sample matrix \(\sqrt{m}L_{mn}^P\). These linear combinations are defined by the elements of the matrix \(R_nA_n\). The orthogonal matrix \(R_n\) is random, while the matrix \(A_n\) is not. However, since the product \(R_nA_n\) is generated independently of \(\sqrt{m}L_{mn}^P\), and since a linear combination of normal random variables remains normally distributed, each column of \(X_{mn}\) will be normally distributed.

In the presence of simulation error in the initial simulation parametric normal ROM simulations remain observationally equivalent to multivariate normal. To demonstrate this we construct a sample \(Z_{mn}\) from \(n\) independent, standard normal distributions, normalise it to have a zero mean and then apply the Gram–Schmidt procedure, which produces a parametric \(L\) matrix \(L_{mn}^P = GS(Z_{mn})\). Then ROM simulations are generated using

\[
X_{mn} = \sqrt{m}L_{mn}^P R_nA_n, \tag{12}
\]

where \(A_n\) is the Cholesky matrix of the target correlation matrix \(C_n\) and \(R_n\) is a random orthogonal matrix. For each type of rotation matrix we generate 10,000 ROM simulations and display their histogram which approximates a marginal distribution of \(X_{mn}\), and plot this against a normal distribution whose parameters are chosen to match the mean and standard deviation of the marginal in question. Each simulation is generated from the same matrix \(L_{mn}^P\) with \(m = 10,000\) and \(n = 10\), and with \(C_{10}\) as shown in Table 3. All densities have mean zero and standard deviation one and we illustrate the 5th marginal distribution of each simulation in Fig. 6.

From Fig. 6 we see that the choice of random orthogonal matrix has negligible effect on the ROM simulated marginals; the marginal distributions of a parametric normal ROM simulation remain normally distributed. To formally classify the marginal distributions of ROM simulations \(X_{mn}\) we need to fully understand the distribution of the fundamental \(L\) matrix \(L_{mn}^P\). In the multivariate normal case this has the unique Haar invariant distribution (see [24]). However, to the best of our knowledge, this classification gives only a partial description; it does not give an explicit representation for marginal distributions.

---

Fig. 6. Histograms for the 5th marginal distribution of a parametric normal ROM simulation. 10,000 observations are used for each simulation. Marginals are compared with scaled normal distributions.
Table 4
Testing the 5th marginal density of parametric normal and Student-t (6 degrees of freedom) ROM simulations ($m = 10,000$ and $n = 10$). Under the null hypothesis, the marginal densities are normal or Student-t respectively.

<table>
<thead>
<tr>
<th></th>
<th>Upper Hessenberg</th>
<th>Cayley</th>
<th>Exponential</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Normality test</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KS stat</td>
<td>0.0057</td>
<td>0.0052</td>
<td>0.0047</td>
</tr>
<tr>
<td>$p$-Value</td>
<td>0.8983</td>
<td>0.9522</td>
<td>0.9801</td>
</tr>
<tr>
<td><strong>Student-t test</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KS stat</td>
<td>0.0312</td>
<td>0.0470</td>
<td>0.0361</td>
</tr>
<tr>
<td>$p$-Value</td>
<td>&lt;0.0001</td>
<td>&lt;0.0001</td>
<td>&lt;0.0001</td>
</tr>
</tbody>
</table>

To identify the true distribution of a given univariate sample $x_1, \ldots, x_m$ we shall use the standard Kolmogorov−Smirnoff test. Consider an empirical distribution $\hat{F}(x)$, calculated from the sample as

$$\hat{F}_m(x) = m^{-1} \sum_{i=1}^{m} I_{x_i \leq x},$$

where $I_{x_i \leq x}$ is the indicator function taking the value one when $x_i \leq x$ and zero otherwise. The Kolmogorov−Smirnoff test statistic for the null hypothesis that the sample distribution in question is $F(x)$, is:

$$D_m = \max_x |F(x) - \hat{F}_m(x)|.$$  

Given a realisation of $D_m$, say $d$, this hypothesis will be rejected if the $p$-value $P(D_m > d)$ is less than a pre-specified significance level. Since the distribution of $D_m$ is complex, the calculation of $p$-values is non-trivial. However, [27] provide a procedure based on a particular representation of $d$. This is implemented in the standard MATLAB function `kstest.m`.

In the upper half of Table 4 we report the test statistics and associated $p$-values corresponding to Kolmogorov−Smirnoff normality tests performed on the 5th marginal distribution of different samples $X_{mn}$. We test three different parametric ROM samples $X_{mn}$, generated from three different types of random rotations. The reported statistics indicate that we should not reject the null hypothesis of normality in all cases. Therefore, even though the normality of $L_{mn}^P$ is not guaranteed in the presence of sampling error, there is empirical evidence to suggest that the

![Histograms for the 5th marginal distribution of a parametric Student-t (6 degrees of freedom) ROM simulation. 10,000 observations are used for each simulation. Marginals are compared with scaled normal distributions.](image-url)
marginals of $X_{mn}$ are normally distributed. Evidence for normality is most strongly supported in the exponential case, where the highest $p$-value is observed.

We now repeat the experiment basing our ROM simulations on multivariate Student-$t$ distributions. The marginal densities are shown in Fig. 7. In contrast to the previous experiment, now the marginal distributions of a ROM simulation depend on the types of random orthogonal matrices used. Whilst symmetry seems relatively unaffected, the distributions simulated using upper Hessenberg ROMs have the highest kurtosis and the exponential ROM simulations have the lowest. Even if we ignore simulation error, so that $\sqrt{mL_P}X_{mn}$ is multivariate Student-$t$ distributed, a linear combination of Student-$t$ random variables need not be Student-$t$ distributed (see [7], for example). Therefore, the marginal distributions of $X_{mn}$ are not necessarily Student-$t$ distributed.

The lower half of Table 4 reports the results of Kolmogorov–Smirnoff tests applied to parametric Student-$t$ ROM simulated samples $X_{mn}$. These samples are constructed as in (12), where $Z_{mn}$ is now a standardised multivariate Student-$t$ sample and the 5th marginal distribution of $X_{mn}$ is compared with a scaled Student-$t$ distributions with six
degrees of freedom. The $p$-values appearing in the row of Table 4 are of the order $10^{-8}$, which strongly suggests that the marginal distributions of a parametric Student-$t$ ROM simulation are not Student-$t$.

### 6.2. Deterministic ROM simulation

ROM simulated marginal densities involving deterministic $L$ matrices are constructed from a large ROM simulated sample by concatenating many samples of the form (12), as explained in [22]. Setting $m = 15$ and $n = 10$, simulations from the Ledermann matrix (2) are repeated until 10,000 observations have been generated.\(^{12}\) The 5th marginal distribution from each ROM simulation is illustrated in Fig. 8. Clearly, the marginal densities are heavy-tailed and

---

\(^{12}\) Given that the dimension of our simulations is $n = 10$, our choice of $m = 15$ gives a Mardia multivariate kurtosis that is slightly greater than normal; see [22] for further details.
skewed, and the size and direction of skew depend on the type of rotation matrix. For example, Cayley rotations induce a high degree of positive marginal skewness.

Other deterministic $L$ matrices generate marginals with very different characteristics. Fig. 9 shows that upper Hessenberg deterministic ROM simulations are fairly symmetric but leptokurtic, especially using Type II and III $L$ matrices. In contrast, Cayley deterministic ROM simulations (Fig. 10) have positive skew when based on the Type I and II $L$ matrices but negative skew when based on the Type III $L$ matrix. Interestingly, using exponential rotations with the Type I $L$ matrix produces marginal densities which are close to being normal. Yet the Type II and III $L$ matrices generate highly non-normal marginals when used in conjunction with exponential ROM simulations; Fig. 11 shows that there is very marked asymmetry in exponential ROM simulations based on the Type II $L$ matrix and a similarly marked, but opposite asymmetry based on the Type III $L$ matrix.

6.3. Hybrid ROM simulation

In the ROM simulation framework it is possible to combine parametric and deterministic methods in many ways, to construct hybrid $L$ matrices that allow for an even greater variation in marginal density characteristics. For example, adding a MVN perturbation to ROM simulations generated by a Ledermann matrix $L_{mn}$ can be achieved via orthogonalizing the augmented matrix

$$A_{m,2n} = (L_{mn}, V_{mn}).$$

![Fig. 12. Histograms for the 5th marginal distribution of a hybrid ROM simulation, using a perturbed Ledermann matrix (perturbation factor $\epsilon = 0.5$). Over 10,000 observations are used for each and simulated marginals are compared with scaled normal distributions.](image-url)
where $V_{mn}$ is a MVN random sample, adjusted to have zero sample mean. Then set $GS(A_{m,2n}) = (\hat{L}_{mn}, \hat{V}_{mn})$. Since $L_{mn}$ is already orthogonal $\hat{L}_{mn} = L_{mn}$, due to the iterative nature of the Gram–Schmidt algorithm. Furthermore, $\hat{V}_{mn}$ will be orthogonal and will satisfy $L_{mn}' \hat{V}_{mn} = 0$. Hence, the hybrid ROM simulations are obtained using the hybrid $L$ matrix

$$L_{mn}' = \frac{1}{\sqrt{1 + \epsilon^2}}(L_{mn} + \epsilon \hat{V}_{mn}).$$

which is rectangular orthogonal for any $\epsilon$.

The construction (13) produces hybrid ROM simulations from any ‘parent’ $L$ matrix with perturbations generated from an elliptical multivariate distribution. The size of the perturbation factor $\epsilon$ controls how closely the hybrid sample characteristics resemble those of samples generated by the parent $L$ matrix. For illustration we generated ROM simulations using a Ledermann $L_{30,10}$ matrix with MVN perturbations. The corresponding marginal densities are shown in Fig. 12. Compared with the densities of Fig. 8 the hybrid ROM simulated densities are more symmetric and have higher central peaks.

### 6.4. Skewness and kurtosis

Figs. 6–12 display a wide variety of shapes that are not always easy to summarize in a few sample statistics. Nevertheless, it is helpful to compare the effects that different rotational matrices have on key statistics such as skewness and kurtosis for a given choice of $L$ matrix. This is summarized in Table 5. A non-zero skewness is most noticeable in deterministic and hybrid ROM simulations, as normal and Student-$t$ distributions are symmetric. With Ledermann, Type I or Type II $L$ matrices an upper Hessenberg rotation induces a negative skew, while Cayley and exponential rotations produce a positive skew in the marginals. The opposite is the case for Type III $L$ matrices. Skew effects are more pronounced for Ledermann and Type II matrix simulations and when Cayley rotations are applied.

Indeed, with Cayley rotations we achieve a significant skewness whatever $L$ matrix used in the ROM simulations. Despite the marked asymmetries that are evident in Fig. 11, exponential rotations have the effect of decreasing the skew, and a significant (and positive) skew is only obtained when using Ledermann and Type II ROM simulations.

Comparing the marginal kurtosis values, three of the four deterministic $L$ matrices have similar kurtosis characteristics – only Type I $L$ matrices have a noticeably lower kurtosis. The hybrid Ledermann-normal ROM simulations produce the most leptokurtic distributions. Even higher kurtosis could be achieved in hybrid ROM simulations by combining any deterministic $L$ matrix (except Type I) with a Student-$t$ $L$ matrix. In most cases the marginal kurtosis of ROM simulations is highest when Cayley rotations are used, and it is almost as high with upper Hessenberg rotations. In all cases the exponential rotation matrix yields simulations with the lowest kurtosis. However, the densities are typically far from having a normal shape, as evidenced by Fig. 11, for instance. In the deterministic case we are also able to increase or reduce the kurtosis by increasing or reducing the number of rows in the $L$ matrix, and concatenating samples.

### 7. Robustness

The results detailed in the previous section were based on the arbitrarily chosen random correlation matrix displayed in Table 3. In this section we examine the robustness of these results to the choice of correlation matrix by repeatedly computing the skewness and kurtosis of a marginal distribution generated in the ROM simulation, each time changing to a different random correlation matrix. These are the steps followed: (i) Select an $L$ matrix $L_{mn}$ and a rotation matrix $R_n$ to use in (4); (ii) Select a $10 \times 10$ random correlation matrix, i.e. a positive matrix with independent, random elements $c_{ij}$ such that $c_{ii} = 1$ and $|c_{ij}| \leq 1$, $i, j = 1, \ldots, 10$; (iii) compute the skewness and kurtosis of a randomly chosen marginal distribution based on 10,000 ROM simulations.

Table 6 reports the results when the computation of marginal skewness and kurtosis is repeated with 1000 different random correlation matrices for each choice of ($L$ matrix, rotation matrix) pair. We report the average skewness (upper part of table) and kurtosis (lower part) over all 1000 ROM-simulated marginal distributions, its standard deviation and use as an indication of robustness the ratio of the mean to the standard deviation reported in the right-hand section of the table. The greater the magnitude of this ratio the more robust are the findings reported in the previous section to the choice of random correlation matrix in Table 3.
Table 6

The average (mean) and standard deviation of marginal skewness and kurtosis based on 10,000 ROM simulations. The mean and standard deviation of these statistics are based on a sample size 1000, where a different random correlation matrix is used to generate each observation on the statistic.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Rotation matrix</th>
<th>Mean Skewness</th>
<th>Std Dev</th>
<th>Mean/Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L matrix ↓</td>
<td>Hess</td>
<td>Cayley</td>
<td>Exp</td>
</tr>
<tr>
<td>Normal</td>
<td>−0.007</td>
<td>0.012</td>
<td>0.002</td>
<td>0.018</td>
</tr>
<tr>
<td>Student−t</td>
<td>0.060</td>
<td>−0.113</td>
<td>−0.067</td>
<td>0.029</td>
</tr>
<tr>
<td>Ledermann</td>
<td>−0.239</td>
<td>0.758</td>
<td>0.503</td>
<td>0.085</td>
</tr>
<tr>
<td>Type I</td>
<td>−0.152</td>
<td>0.450</td>
<td>0.276</td>
<td>0.042</td>
</tr>
<tr>
<td>Type II</td>
<td>−0.207</td>
<td>0.750</td>
<td>0.473</td>
<td>0.067</td>
</tr>
<tr>
<td>Type III</td>
<td>0.200</td>
<td>−0.714</td>
<td>−0.491</td>
<td>0.065</td>
</tr>
<tr>
<td>Hybrid</td>
<td>−0.200</td>
<td>0.830</td>
<td>0.644</td>
<td>0.075</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Rotation matrix</th>
<th>Mean Kurtosis</th>
<th>Std Dev</th>
<th>Mean/Std Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>L matrix ↓</td>
<td>Hess</td>
<td>Cayley</td>
<td>Exp</td>
</tr>
<tr>
<td>Normal</td>
<td>2.972</td>
<td>2.976</td>
<td>2.984</td>
<td>0.038</td>
</tr>
<tr>
<td>Student−t</td>
<td>4.434</td>
<td>4.703</td>
<td>3.923</td>
<td>0.412</td>
</tr>
<tr>
<td>Ledermann</td>
<td>4.652</td>
<td>4.532</td>
<td>3.248</td>
<td>0.342</td>
</tr>
<tr>
<td>Type I</td>
<td>3.465</td>
<td>3.581</td>
<td>2.790</td>
<td>0.148</td>
</tr>
<tr>
<td>Type II</td>
<td>4.575</td>
<td>4.562</td>
<td>3.385</td>
<td>0.331</td>
</tr>
<tr>
<td>Type III</td>
<td>4.571</td>
<td>4.517</td>
<td>3.247</td>
<td>0.330</td>
</tr>
<tr>
<td>Hybrid</td>
<td>6.657</td>
<td>6.458</td>
<td>4.778</td>
<td>0.436</td>
</tr>
</tbody>
</table>

Although the skewness results are less robust than the kurtosis results in Table 6, in most cases the standard deviation is still considerably less than the sample average. The kurtosis results in the lower part of the table are particularly robust for the normal parametric L matrices, and otherwise the ratio of the mean to standard deviation is greater than 10 with only one exception. By contrast, the near-zero skewness obtained using normal parametric ROM simulations is more sensitive to the correlation matrix than the skewness based on other types of L matrix. Also, while exponential rotation matrices produce the least robust skewness results, they have the most robust kurtosis. That is, the mean-standard deviation ratio for kurtosis based on exponential rotation matrices is greater than for Cayley or Hessenberg rotation matrices, for every type of L matrix chosen. As in the previous section’s results, the average exponential ROM marginal kurtosis is consistently lower than the average kurtosis based on Cayley or Hessenberg rotation matrices, except when a normal parametric L matrix is used in which case all kurtosis statistics are very close to three.

Table 7

Computation time of normal Monte Carlo relative to parametric normal ROM simulations. The column labels 10, 50 and 100 correspond to the dimension of the simulations. All simulated arrays have 10,000 rows. The top part of the table quotes the computation time for a non-exact covariance MC simulation, relative to the exact covariance ROM simulation. The lower part of the table quotes the computation time for an exact covariance MC simulation, relative to the exact covariance ROM simulation.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Rotation matrix</th>
<th>10</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-exact Monte Carlo</td>
<td>Upper Hessenberg</td>
<td>1.7435</td>
<td>1.2211</td>
<td>1.2194</td>
</tr>
<tr>
<td></td>
<td>Cayley</td>
<td>0.4884</td>
<td>0.2952</td>
<td>0.1555</td>
</tr>
<tr>
<td></td>
<td>Exponential</td>
<td>1.1773</td>
<td>1.3122</td>
<td>1.0475</td>
</tr>
<tr>
<td>Exact Monte Carlo</td>
<td>Upper Hessenberg</td>
<td>7.0843</td>
<td>39.2779</td>
<td>67.6795</td>
</tr>
<tr>
<td></td>
<td>Cayley</td>
<td>1.3921</td>
<td>8.3437</td>
<td>8.6106</td>
</tr>
<tr>
<td></td>
<td>Exponential</td>
<td>3.8273</td>
<td>35.9223</td>
<td>59.2859</td>
</tr>
</tbody>
</table>
Table 8
Computation time of Student-$t$ Monte Carlo relative to parametric Student-$t$ ROM simulations. The column labels 10, 50 and 100 correspond to the dimension of the simulations. All simulated arrays have 10,000 rows. The top part of the table quotes the computation time for a non-exact covariance MC simulation, relative to the exact covariance ROM simulation. The lower part of the table quotes the computation time for an exact covariance MC simulation, relative to the exact covariance ROM simulation.

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>50</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-exact Monte Carlo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Upper Hessenberg</td>
<td>549.19</td>
<td>757.20</td>
<td>575.83</td>
</tr>
<tr>
<td>Cayley</td>
<td>89.65</td>
<td>149.57</td>
<td>81.98</td>
</tr>
<tr>
<td>Exponential</td>
<td>252.65</td>
<td>760.65</td>
<td>504.16</td>
</tr>
<tr>
<td>Exact Monte Carlo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Upper Hessenberg</td>
<td>600.57</td>
<td>822.56</td>
<td>867.73</td>
</tr>
<tr>
<td>Cayley</td>
<td>107.69</td>
<td>165.82</td>
<td>93.66</td>
</tr>
<tr>
<td>Exponential</td>
<td>275.72</td>
<td>713.98</td>
<td>641.04</td>
</tr>
</tbody>
</table>

8. Computational time

Monte Carlo methods are computationally complex since they involve inverse cumulative distribution functions, which are not always available analytically. Fortunately, there are efficient methods for approximating these functions in several cases (e.g. with a multivariate normal distribution) but whenever the inverse of a distribution cannot be approximated accurately or efficiently Monte Carlo simulation can become an arduous task.

ROM simulations rely on matrix multiplication rather than distributional sampling so they are computationally more straightforward. The advantage is that matrix multiplication is fast and easy to implement, allowing many samples to be generated as random transforms of a single orthogonal data set. Yet, however efficient this method may be, we should remind ourselves that samples are linked by (random) linear transformations. Introducing a random permutation matrix $Q_m$ in (4) will destroy the correlation between ROM-simulated random samples, but they are not independent in the sense that Monte Carlo random samples are. Another important distinction is that in many instances ROM simulation is a non-parametric or semi-parametric method. That is, apart from the multivariate normal case, the distribution of ROM simulated samples is often unknown. Monte Carlo simulation is always parametric and if the sample size is large enough the empirical distribution of the sample will closely resemble the target distribution.

We now present an experiment that starts with the generation of a large standard multivariate normal sample $Z_{mn}$ using Monte Carlo techniques. To prepare our ROM simulations we set $L_{mn} = GS(Z_{mn} - 1_m \tilde{z}_n)$. An exact mean-covariance sample can be constructed as $X_{mn}^{\text{ROM}} = \sqrt{m}L_{mn}$. The standard Monte Carlo sample is simply $X_{mn}^{\text{MC}} = Z_{mn}$.

Now suppose we want to generate further multivariate samples, using our existing $Z_{mn}$ when possible. To generate another exact sample we only need to generate another $n \times n$ random matrix. To generate another (non-exact) Monte Carlo sample we need to generate another $m \times n$ array of normal variates.

Thus, we compare the computational speed of the following operations: 1. Generate a random orthogonal $R_n$ and form $X_{mn}^{\text{ROM}} = \sqrt{m}L_{mn}R_n$; 2. Generate a new Monte Carlo sample $Z_{mn}$, and set $X_{mn}^{\text{MC}} = Z_{mn}$. Table 7 compares the times taken to perform the above with $m = 10$, $000$ and $n = 10$, $50$ and $100$, and where the random orthogonal matrices $R_{mn}$ are formed in three different ways. Computations were carried out in Matlab on an Intel(R) Xeon(R) CPU, 2.67 GHz, with 3.00 GB of RAM. The figures quoted are the ratio of the time taken for operation 2 relative to operation 1. However, we are not really comparing like with like here, as operation 1 is an exact simulation method whereas operation 2 is not. Therefore, in the lower half of Table 7 we present the time taken to generate a new Monte Carlo sample $Z_{mn}$, letting $X_{mn}^{\text{MC}} = GS(Z_{mn} - 1_m \tilde{z}_n)$, and compute the time taken relative to the ROM simulation operation 1.

The values in the top half of Table 7 show that generating multivariate samples using Upper Hessenberg or exponential rotations is faster than standard Monte Carlo techniques. For example, for a simulation with dimension $n = 10$, upper Hessenberg ROM simulations are $1.7435$ times faster than Monte Carlo simulations. As discussed in Section 6.1, the original Monte Carlo simulation error carries over into the ROM simulations. So both samples have simulation error, but only the ROM samples have exact means and covariance. Exact Monte Carlo simulations are even slower to generate relative to ROM simulations because the Gram–Schmidt (or other) orthogonalization procedure must be used at every
simulation. In contrast, using the ROM simulation framework we only need to apply Gram–Schmidt once, to form an $L$ matrix. After this initial orthogonalization, infinitely many exact samples can be generated from this same $L$ matrix.

Under more complex distributional assumptions Monte Carlo simulation becomes even more time consuming. For illustration we repeat the above experiment using a multivariate Student-$t$ distribution with 6 degrees of freedom. The relative computational times given in Table 8 show that ROM simulation can be several hundred times faster than Student-$t$ Monte Carlo simulation, depending on the system dimension and the rotational matrix used. However ROM simulated samples based on parametric Student-$t$ $L$ matrices are not necessarily Student-$t$ distributed. Again, the upper Hessenberg ROM simulations were fastest and Cayley ROM simulations are the slowest. For simulating high dimensional systems (with $n = 50$ or $n = 100$) exponential ROM simulations are almost as fast as upper Hessenberg.

9. Summary and conclusions

ROM simulation is a new technique for generating random samples, which encompasses Monte Carlo and historical simulation as special cases. The fundamental idea is to capture the characteristics of simulations via an $L$ matrix, and to multiply this matrix by random permutation, reflection and/or rotation matrices. It has the advantage of producing very fast simulations with exact mean and covariance. In data-specific and parametric ROM simulation the higher multivariate moments are identical to those of an observed sample (empirically observed or parametrically simulated).

In deterministic ROM simulation the $L$ matrix is constructed without reference to an observed sample. Instead, the matrix has parameters that are chosen to target some pre-assigned values for skewness and kurtosis. Exact matching to skewness and kurtosis is not always possible due to integer parameter constraints, but calibration errors are very small.

This paper provides further insight to the properties of ROM simulation, beyond those introduced in [22]. When the sample size parameter of deterministic ROM simulation is small, or when there are only a few historical observations in the variables, the ROM simulations may be concatenated to obtain much larger samples. We demonstrate algebraically and empirically that the Mardia skewness will decrease at a rate approximately inversely proportional to the number of concatenations. However, the mean, covariance and kurtosis are left invariant by concatenation.

We investigate the time-series properties of data-specific ROM simulations using real financial data, and of parametric ROM simulations derived from a Markov-switching GARCH model. Here the pre-multiplication of the $L$ matrix by a cyclic permutation matrix plays a critical role. Cyclic ROM simulations may have useful applications to the reduction of data-snooping bias, amongst other econometric problems.

We then explore how different types of random rotations produce samples with different marginal characteristics, and we also demonstrate that ROM simulation is very much faster than parametric Monte Carlo simulation particularly when exact means and covariance matrices are required for each simulation. Upper Hessenberg ROM simulation is the fastest in small dimensional systems, but in large systems exponential ROM simulation is almost equally fast.

The choice of rotational matrix then depends on the sample characteristics that are required. Exponential, Cayley and Hessenberg rotation matrices produce ROM simulations that are far from normal, except when a normal parametric $L$ matrix is used. The exponential ROM simulation has the lowest kurtosis, but still it is significantly greater than three. Hessenberg and exponential ROM simulated samples have opposite skewness characteristics: with Type I or Type II $L$ matrices, Hessenberg ROM simulations have negative skew and exponential ROM simulations have positive skew; the opposite is the case with Type III $L$ matrices. Cayley ROM simulation is slower than Hessenberg or exponential ROM simulation, but it is still about 100 times faster than Student-$t$ Monte Carlo. The kurtosis characteristics of Cayley ROM simulations are similar to those of Hessenberg ROM simulations. The main advantage of using Cayley ROM simulation is that they produce large values for the sample skewness, which is positive with Type I or Type II $L$ matrices, and negative with Type III $L$ matrices.

An Appendix A explains how to calibrate deterministic $L$ matrices to target any values for skewness and kurtosis that we choose. An illustration of the calibration of Type I $L$ matrices to time series of data on the 45 Morgan Stanley Country Indices shows that calibration errors are small.

ROM simulation has great potential for financial applications. Interesting problems might include: the application of data-specific ROM simulation to assess model risk for problems that are typically resolved using a single historically observed sample, such as Value-at-Risk; stress testing portfolios using deterministic ROM simulation, where $L$ matrices target extreme values for skewness and kurtosis; performance evaluation of optimal portfolios, and/or of econometric models, where cyclic ROM simulations are applied to reduce the data-snooping bias; and speeding up option pricing and hedging computations in the non-affine model class, and/or for risk-neutral valuation of path-dependent pay-offs.
Fig. 13. Grid for targeting skewness and kurtosis using $L^k$-matrices of Type I with dimension $n = 45$, $\hat{\tau} = 1386$ and $\hat{\kappa} = 4111$. The parameters $m$ and $k$ are selected as the integer coordinate pair closest to the coordinates at the minimum point.

under affine as well as non-affine models. ROM simulation can be applied to resolve any problem where Monte Carlo simulation is typically applied, but it is much faster than Monte Carlo. Hence, ROM simulation might also be applied to data-prediction problems in econometrics, climate, hydrology and other non-financial disciplines.

Acknowledgments

We would like to thank the editor and two anonymous referees for some very useful comments which have greatly improved the paper.

Appendix A. Calibration of deterministic $L$-matrices

Unconstrained by distributional assumptions or historical data, deterministic $L$-matrices are simply characterised by their dimension $m \times n$ and a moment matching parameter $k$. Whilst $n$ is fixed by the number of variables the integers $m$ and $k$ are free to change, and we here consider how they may be calibrated to target the multivariate skewness $\hat{\tau}$ and kurtosis $\hat{\kappa}$. So, with $n$ fixed, we set up an objective function such as:

$$ fn(m, k; \hat{\tau}, \hat{\kappa}) = \left( \frac{\tau(L_{mn}^k)}{\hat{\tau}} - \hat{\tau} \right)^2 + \left( \frac{\kappa_M(L_{mn}^k)}{\hat{\kappa}} - \hat{\kappa} \right)^2. $$ (14)

An integer pair $(m^*, k^*)$ which minimises $f_n(m, k; \hat{\tau}, \hat{\kappa})$ can be found via any derivative free numerical optimisation scheme. However, we found it more time effective to construct a skewness and kurtosis grid, such as that shown Fig. 13, which gives the value of (14) for different values of $m$ and $k$. That is, we calculate $\tau(L_{mn}^k)$ and $\kappa_M(L_{mn}^k)$ for a large range of integer pairs $(m, k)$ and thereby form a grid of $f_n(m, k; \hat{\tau}, \hat{\kappa})$ for some fixed $n$, any given pair $(\hat{\tau}, \hat{\kappa})$.

The surface shown in Fig. 13 was based on $n = 45$, $\hat{\tau} = 1386$ and $\hat{\kappa} = 4111$. These values have been chosen for illustration. They were calculated from the 500 daily returns on the 45 Morgan Stanley Country Indices (MSCI) leading up to 19 September 2008, which was at the peak of the banking crisis.\footnote{See http://www.mscibarra.com for more details about these indices.}

The minimum point on the grid occurs at the point $(183, 35)$, when the objective function takes the value 0.0076. With these parameters, our ROM simulated samples have a skewness of 1390 and a kurtosis equal to 4141, which are
close but not exactly equal to the historical targets. This approximation error is typical and it arises because $m$ and $k$ are constrained to be integers.

Next, we report the calibration error, as a percentage of the underlying parameter (Mardia skewness, or kurtosis) as the multivariate sample of 500 returns on the 45 county indices is rolled over time. The total data period starts on 29 August 1997 and ends on 11 January 2010, and has over 3000 daily returns on each index. Starting with the 500 returns from August 1997 to August 1999, we compute the Mardia skewness and kurtosis, set these as the target values $\hat{\tau}$ and $\hat{\kappa}$ and find $m$ and $k$ minimize the objective (14). Then the period is rolled forward one day, and the calibration is repeated, until the entire sample is exhausted. **Fig. 14** depicts the time series of target skewness (in black), the calibrated skewness value (in blue) and the percentage calibration error (in red). **Fig. 15** depicts the corresponding series for kurtosis. Both calibration errors (shown on the right-hand scale) are never greater than 5%, and typically lie in the region of $\pm 2\%$. This supports our earlier statement that small calibration errors are typical and arise from the integer constraints on $m$ and $k$.

![Fig. 14. Time series of target skewness (in black), the calibrated skewness value (in blue) and the percentage calibration error (in red), based on the MSCI data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)](image1)

![Fig. 15. Time series of target skewness (in black), the calibrated skewness value (in blue) and the percentage calibration error (in red), based on the MSCI data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of the article.)](image2)
References


