BIT
2000, Vol. 40, No. 4, pp. 640–651

0006-3835/00/4004-0640 $15.00

NUMERICALLY STABLE GENERATION OF CORRELATION MATRICES AND THEIR FACTORS [*]

PHILIP I. DAVIES[†] and NICHOLAS J. HIGHAM[‡]

Department of Mathematics, University of Manchester, Manchester, M13 9PL, England
email: ieuan@ma.man.ac.uk, higham@ma.man.ac.uk

Abstract.

Correlation matrices—symmetric positive semidefinite matrices with unit diagonal—are important in statistics and in numerical linear algebra. For simulation and testing it is desirable to be able to generate random correlation matrices with specified eigenvalues (which must be nonnegative and sum to the dimension of the matrix). A popular algorithm of Bendel and Mickey takes a matrix having the specified eigenvalues and uses a finite sequence of Givens rotations to introduce 1s on the diagonal. We give improved formulae for computing the rotations and prove that the resulting algorithm is numerically stable. We show by example that the formulae originally proposed, which are used in certain existing Fortran implementations, can lead to serious instability. We also show how to modify the algorithm to generate a rectangular matrix with columns of unit 2-norm. Such a matrix represents a correlation matrix in factored form, which can be preferable to representing the matrix itself, for example when the correlation matrix is nearly singular to working precision.

Key words: Random correlation matrix, Bendel–Mickey algorithm, eigenvalues, singular value decomposition, test matrices, forward error bounds, relative error bounds, IMSL, NAG Library, Jacobi method.

AMS subject classification: 65F15.

1 Introduction.

An important class of symmetric positive semidefinite matrices is those with unit diagonal. They arise in statistics as matrices of correlation coefficients and are known as correlation matrices [16], [25, p. 24]. They also play an important role in numerical analysis, because of an approximate optimality property. Let $A \in \mathbb{R}^{n\times n}$ be symmetric positive definite and write

$$A = DHD, \qquad D = \mathrm{diag}(A)^{1/2}, \tag{1.1}$$

[*] Received November 1999. Revised May 2000. Communicated by Lars Eldén.

[†] Supported by an Engineering and Physical Sciences Research Council CASE Ph.D. Studentship with NAG (Oxford) as the cooperating body.

[‡] Supported by Engineering and Physical Sciences Research Council grant GR/L76532 and by a Royal Society Leverhulme Trust Senior Research Fellowship.

so that H has unit diagonal and hence is a correlation matrix. A result of van der Sluis [29] states that

$$\kappa_2(H) \le n \min_{F \text{ diagonal}} \kappa_2(FAF), \tag{1.2}$$

where $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2$. Thus H comes within a factor n of minimizing the 2-norm condition number over all two-sided diagonal scalings of A. This property accounts for the appearance of correlation matrices in several contexts in numerical analysis, three of which we briefly explain.

Cholesky factorization. Standard error analysis shows that when Cholesky factorization is used to solve a symmetric positive definite linear system $Ax = b$ in floating point arithmetic, the relative error $\|x - \widehat{x}\|_2/\|x\|_2$ of the computed solution $\widehat{x}$ is bounded by a multiple of $\kappa_2(A)u$, where u is the unit roundoff. A more refined analysis ([5], [15, Sec. 10.1], [30]) shows that $\|D(x - \widehat{x})\|_2/\|Dx\|_2$ is bounded by a multiple of $\kappa_2(H)u$, with D and H as defined in (1.1). If A is "artificially ill conditioned", in the sense that H is well conditioned and D has diagonal elements of widely varying magnitude, then the latter bound can guarantee much better relative accuracy in $\widehat{x}$ than the original one. The refined analysis also provides a condition for Cholesky factorization to succeed in floating point arithmetic (that is, for no square roots of negative quantities or divisions by zero to occur) involving an upper bound on $\kappa_2(H)u$ rather than the more usual $\kappa_2(A)u$.

Preconditioning. The simplest preconditioner for iterative methods for solving a symmetric positive definite linear system $Ax = b$ is the Jacobi preconditioner [1, 13], which uses D in (1.1) to transform to a linear system involving the correlation matrix H. The motivation is the result (1.2), since one of the aims of preconditioning is to reduce the condition number $\kappa_2(A)$. For some matrices arising in PDE applications, D in (1.1) is an optimal scaling: Forsythe and Straus [11] show that $\kappa_2(H)$ is minimal if A is symmetric positive definite with property A (that is, there exists a permutation matrix P such that PAP^T can be expressed as a block 2×2 matrix whose $(1, 1)$ and $(2, 2)$ blocks are diagonal). Thus, for example, any symmetric positive definite block tridiagonal matrix whose diagonal blocks are identities is optimally scaled. For a summary of more general scaling results of this type see [13, Sec. 10.5].

Jacobi methods. Jacobi's method for finding the eigensystem of a symmetric matrix A is an old method for which attractive high accuracy properties have been proved in recent years. Demmel and Veselić [7] (see also Mathias [23]) show that when Jacobi's method is applied to a symmetric positive definite matrix A, using a suitable stopping criterion, the computed eigenvalues satisfy relative error bounds proportional to $\kappa_2(H)u$ rather than the quantity $\kappa_2(A)u$ that would be expected for a normwise backward stable method.

For testing theory and algorithms in statistics and these numerical analysis applications, and for simulations in signal processing [16], it is desirable to be able to generate correlation matrices with a specified eigenvalue distribution or 2-norm condition number. One approach, used by Demmel and Veselić [7], is to generate a random symmetric positive definite matrix with specified eigenvalues by any existing technique and then scale it to produce unit diagonal. However, the act of scaling can change the condition number by an arbitrary amount. Certain special matrices are available with the desired properties. For example, the Kac–Murdock–Szego Toeplitz matrix with (i, j) element $\rho^{|i-j|}$ has unit diagonal and is positive semidefinite if $|\rho| \le 1$. Choosing ρ to achieve a desired condition number is possible but nontrivial, as the eigenvalues are known only in terms of the roots of a certain nonlinear equation [28]. For $C \in \mathbb{R}^{m\times n}$ with $\|C\|_2 < 1$, the matrix

$$A = \begin{bmatrix} I_m & C \\ C^T & I_n \end{bmatrix}$$

(which has property A and so is optimally scaled) is easily shown to be positive definite with $\kappa_2(A) = (1 + \|C\|_2)/(1 - \|C\|_2)$, but A has many zero elements so is not necessarily a good test matrix for the Cholesky or Jacobi methods.

In Section 3 we describe an algorithm of Bendel and Mickey for generating a random correlation matrix with specified spectrum. We give new formulae for the rotations used in the algorithm. In Section 4 we prove that the Bendel–Mickey algorithm with the new formulae is numerically stable and that it is safe to set the computed diagonal to unity. We show that the original formulae for the rotations, which are used in current software, can lead to serious instability. When the desired condition number approaches the reciprocal of the unit roundoff the computed correlation matrix can be indefinite. In this case it is better to represent the correlation matrix in factored form, X^TX, where X is $m \times n$ with columns of unit norm. In Section 5 we give a new, one-sided version of the Bendel–Mickey algorithm for generating an $m \times n$ factor X with columns of unit norm and specified singular values, with X optionally the upper triangular Cholesky factor of the corresponding correlation matrix.

We begin, in the next section, by briefly reviewing some necessary background theory.

2 Theory.

Constraints on the spectrum of a correlation matrix can be deduced from the following standard majorization result [17, Theorems. 4.3.26, 4.3.32] (see [20] for a recent geometric proof).

THEOREM 2.1. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric. A necessary and sufficient condition for A to have eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ and diagonal elements $\alpha_1 \le \alpha_2 \le \cdots \le \alpha_n$ (in any order along the diagonal) is that*

$$\sum_{i=1}^{k} \lambda_i \le \sum_{i=1}^{k} \alpha_i, \qquad k = 1, \dots, n, \tag{2.1}$$

with equality for $k = n$.

Note that the requirement of equality in (2.1) for $k = n$ follows trivially from the fact that $\operatorname{trace}(A) = \sum_i \alpha_i = \sum_i \lambda_i$. For a correlation matrix, $\alpha_i \equiv 1$

and $\lambda_i \geq 0$, and the theorem implies that any set of nonnegative eigenvalues summing to n is possible.

A natural question, which does not seem to have been considered before, is whether it is possible to generate a banded correlation matrix with specified nonnegative spectrum summing to n. Recall that A has bandwidth p if $a_{ij} = 0$ for $|i - j| > p$. The next result gives a negative answer to this question for the narrow bandwidth case that is of most practical interest.

Theorem 2.2. *Not every set of nonnegative scalars $\lambda_1, \ldots, \lambda_n$ summing to n can be the eigenvalues of a correlation matrix of bandwidth p if $p < (n-1)/2$.*

Proof. Let $A \in \mathbb{R}^{n\times n}$ be a correlation matrix of bandwidth p. Because A is positive semidefinite with unit diagonal, its off-diagonal elements are bounded in magnitude by 1. The condition on p implies that every row of A has less than $n-1$ nonzero elements. Gershgorin's theorem therefore implies that every eigenvalue λ of A satisfies $\lambda < n$. Thus $\lambda_1 = n$ and $\lambda_2 = \cdots = \lambda_n = 0$ is an example of a set of eigenvalues that is not achievable. □

Our interest is now in transforming a given symmetric positive semidefinite matrix with trace n into a correlation matrix, while preserving the eigenvalues. That such a transformation is possible follows from Theorem 2.1 and is also a corollary of the following more general result that applies to nonsymmetric matrices. This result has a long history in which it appears in various forms; see, in particular, [10], [18, Theorem. 1.3.4], [22], [26]. We give a proof, since it suggests an algorithm.

Theorem 2.3. *If $A \in \mathbb{C}^{n\times n}$ has trace n then there exists a unitary U such that U^*AU has unit diagonal. If A is real or Hermitian then U can be taken to be real.*

Proof. Since the field of values $F(A) = \{\, x^*Ax : x^*x = 1,\ x \in \mathbb{C}^n \,\}$ is a convex set containing the eigenvalues of A,

$$1 = \operatorname{trace}(A)/n = (\lambda_1 + \cdots + \lambda_n)/n \in F(A).$$

Hence there exists $x \in \mathbb{C}^n$ such that

$$x^*Ax = 1, \qquad x^*x = 1. \tag{2.2}$$

Choose $\widetilde{Q}$ so that $[x\ \widetilde{Q}]$ is unitary and form

$$\begin{bmatrix} x^* \\ \widetilde{Q}^* \end{bmatrix} A \begin{bmatrix} x & \widetilde{Q} \end{bmatrix} = \begin{bmatrix} 1 & x^*A\widetilde{Q} \\ \widetilde{Q}^*Ax & \widetilde{Q}^*A\widetilde{Q} \end{bmatrix}.$$

Taking the trace, we find that the matrix $\widetilde{Q}^*A\widetilde{Q} \in \mathbb{C}^{(n-1)\times(n-1)}$ has trace $n-1$ and so the first part of the result follows by induction, since the case $n = 1$ is trivial. If A is real or Hermitian we have to show that $x = u + iv$ in (2.2) can be taken to be real. We can assume that $u \neq 0$, because if $u = 0$ then $x/i = v$ is a real vector satisfying (2.2). From the real parts of (2.2) we have

$$u^*Au + v^*Av = 1, \qquad u^*u + v^*v = 1.$$

Subtracting these two equations gives

$$u^*(A-I)u = -v^*(A-I)v. \tag{2.3}$$

Defining the real vector $x(t) = u+tv$, we now try to find t such that $x(t)^*Ax(t) = x(t)^*x(t)$. This equation can be written

$$u^*Au + t(u^*Av + v^*Au) + t^2v^*Av = u^*u + t(u^*v + v^*u) + t^2v^*v,$$

or, using (2.3),

$$at^2 + bt - a = 0,$$

where $a = v^*(A-I)v$ and $b = u^*(A-I)v + v^*(A-I)u$. When A is real or Hermitian both a and b are real and we have the real solutions

$$t = \frac{-b \pm \sqrt{b^2+4a^2}}{2a},$$

or $t = 0$ if $a = 0$. Since $u \neq 0$, $x(t) = u + tv$ is nonzero for one of the solutions t and so $x(t)/\|x(t)\|_2$ is a real vector satisfying (2.2). □

An algorithm based on the construction in this proof is given by Marsaglia and Olkin [22]. It rests on the ability to construct a vector x satisfying (2.2). Finding x is not straightforward, and an iterative technique is used in [22]. Unfortunately, no upper bound on the number of iterations is available, although the average behaviour is stated to be satisfactory when A is a diagonal matrix of random eigenvalues. We therefore turn our attention in the next section to a method of Bendel and Mickey, which is easy to implement and has a known and satisfactory computational cost.

3 The Bendel–Mickey algorithm.

Given a symmetric positive semidefinite matrix $A \in \mathbb{R}^{n\times n}$ with trace$(A) = n$ the Bendel–Mickey algorithm [2] transforms A into a correlation matrix in $n-1$ steps, each of which consists of applying an orthogonal similarity transformation in an (i,j) coordinate plane to introduce a 1 in the (i,i) position. (This method is outlined by Golub and Van Loan in Problems[1] 8.4.1 and 8.4.2 of [12].) It suffices to describe the first stage. If A does not have unit diagonal then, since trace$(A) = n$, there exist i and j with $i < j$ such that $a_{ii} < 1 < a_{jj}$ or $a_{ii} > 1 > a_{jj}$. We apply a Givens rotation to obtain

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}^T \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} 1 & \widetilde{a}_{ij} \\ \widetilde{a}_{ij} & \widetilde{a}_{jj} \end{bmatrix}, \qquad c^2+s^2=1. \tag{3.1}$$

Equating (1,1) elements yields the equation

$$c^2a_{ii} - 2sca_{ij} + s^2a_{jj} = 1. \tag{3.2}$$

[1] In the 1989 second edition of [12] these problems are numbered 8.5.3 and 8.5.4.

In [2], and in the Fortran implementation in [21], this equation is solved by converting it into a quadratic in $\cos 2\theta$ (where $c = \cos\theta$). However, it is better in terms of efficiency and numerical stability to express (3.2) as a quadratic in $t = s/c$,

$$(a_{jj} - 1)t^2 - 2ta_{ij} + a_{ii} - 1 = 0, \tag{3.3}$$

from which

$$t = \frac{a_{ij} \pm \sqrt{a_{ij}^2 - (a_{ii} - 1)(a_{jj} - 1)}}{a_{jj} - 1}. \tag{3.4}$$

Note that the choice of i and j ensures that $(a_{ii} - 1)(a_{jj} - 1) < 0$ and hence that t is real; moreover, the argument of the square root is evaluated accurately as there is no damaging cancellation. We take the sign in (3.4) to avoid cancellation, obtaining the other root, if necessary, from the fact that the product of the roots is $(a_{ii} - 1)/(a_{jj} - 1)$. Then we recover

$$c = \frac{1}{\sqrt{1 + t^2}}, \qquad s = ct. \tag{3.5}$$

We could, alternatively, use a Householder reflection, as suggested by Marsaglia and Olkin [22]. However, any 2×2 Householder reflection has the form

$$\begin{bmatrix} -c & s \\ s & c \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix},$$

from which it is easily seen that applying a Householder reflection is equivalent, up to signs, to applying a Givens rotation.

If we start with a diagonal A this approach does not necessarily produce a full matrix (that is, some of the correlations may be zero). To ensure that a full matrix is produced and to add randomness we begin by applying a random orthogonal similarity transformation, for which the natural distribution is the Haar distribution [27].

Algorithm 3.1 (Bendel and Mickey). Given nonnegative scalars $\lambda_1, \dots, \lambda_n$ summing to n, this algorithm produces a random correlation matrix having eigenvalues $\lambda_1, \dots, \lambda_n$.

```
1. Form a random orthogonal matrix $U \in \mathbb{R}^{n\times n}$ from the Haar distribution.
   Let $A = U\,\mathrm{diag}(\lambda_i)U^T$.
2. while some $a_{ii} \neq 1$
        Find indices $i$ and $j$ with $i < j$ so that $a_{ii} < 1 < a_{jj}$ or $a_{ii} > 1 > a_{jj}$.
        Apply an orthogonal similarity transformation comprising
        a Givens rotation in the $(i, j)$ plane to set $a_{ii} = 1$,
        as described by (3.1), (3.4) and (3.5).
   end
```

The matrix U in Algorithm 3.1 can be formed by the method of Stewart [27] (see also [15, Section 26.3]), which constructs it as a product of $n - 1$ random Householder transformations and a diagonal matrix of ±1s.

Algorithm 3.1 has several attractive features. First, it has a high degree of randomness: the first stage produces a matrix with random eigenvectors from a standard distribution, and the indices in the loop and the root of (3.3) can be selected randomly, too, if desired. Second, the algorithm is of known and reasonable cost. The initial formation of A requires $7n^3/3$ flops using Stewart's method. The while loop requires at most $n-1$ iterations so the second stage costs at most $12n^2$ flops, which is negligible compared with the first stage. Finally, software for the first stage is readily available, in MATLAB as `gallery('randsvd',...)`, and in Fortran 77 in directory `lapack/testing/matgen` of the LAPACK distribution (see [6] for documentation), so the algorithm is very easy to implement. Algorithm 3.1 is implemented in MATLAB 6 as `gallery('randcorr',...)`.

The statistical distribution of the matrices produced by Algorithm 3.1 is not well understood; see Holmes [16] for details of what is known.

Finally, we note that an algorithm similar to Algorithm 3.1 for constructing a symmetric matrix with given eigenvalues and diagonal elements satisfying the majorization condition (2.1) is given by Chan and Li [4], with a Fortran implementation in [3]. As Ikramov [19] has noted, the same algorithm is also given, without reference to [4], by Zha and Zhang [31].

4 Error analysis and implementation issues.

Now we consider the behaviour of Algorithm 3.1 in floating point arithmetic and compare it with the existing version of the Bendel and Mickey algorithm in [2], [21].

We define

$$\tilde{\gamma}_k = \frac{cku}{1-cku},$$

in which c denotes a small integer constant whose exact value is unimportant.

LEMMA 4.1. *Let $\widehat{A}_i$ denote the computed matrix at the end of stage i in Algorithm 3.1, for $i = 1, 2$. Then*

$$\begin{aligned} \widehat{A}_1 &= Q_1^T(\operatorname{diag}(\lambda_i) + \Delta A_1)Q_1, \qquad \|\Delta A_1\|_F \le \tilde{\gamma}_{n^2}\|\operatorname{diag}(\lambda_i)\|_F, \\ \widehat{A}_2 &= Q_2^T(\widehat{A}_1 + \Delta A_2)Q_2, \qquad \|\Delta A_2\|_F \le \tilde{\gamma}_n\|\widehat{A}_1\|_F, \end{aligned}$$

where Q_1 and Q_2 are orthogonal. Consequently, the output of the algorithm is a matrix $\widehat{A}$ satisfying

$$\widehat{A} = Q^T(\operatorname{diag}(\lambda_i) + \Delta)Q, \qquad \|\Delta\|_F \le \tilde{\gamma}_{n^2}\|\operatorname{diag}(\lambda_i)\|_F,$$

where Q is orthogonal.

PROOF. The result for stage 1 follows from standard error analysis of Householder transformations [15, Sec. 18.3]. Similarly, for stage 2 we can adapt standard error analysis of Givens transformations [15, Sec. 18.5]. □

The lemma shows that most of the error in Algorithm 3.1 comes, as does most of the work, from stage 1. It follows from standard perturbation theory that the eigenvalues of $\widehat{A}$ differ from the desired eigenvalues λ_i by absolute amounts at most $\|\Delta\|_F \le \sqrt{n}\tilde{\gamma}_{n^2} \max_i \lambda_i$. If $\max(\lambda_i)/\min(\lambda_i) > n^{-1/2}\tilde{\gamma}_{n^2}^{-1}$ then $\widehat{A}$ can be indefinite and so fail to be a correlation matrix. In this situation it is preferable, if the application allows it, to generate A in factored form, as described in the next section.

The computed $\widehat{A}$ from Algorithm 3.1 will not usually have diagonal elements exactly equal to 1, because of rounding errors. The question arises of whether it is safe to set these elements to 1. Consider the general step in stage 2, which is designed to set a_{ii} to unity as described by (3.1), and suppose that $a_{ii} < 1 < a_{jj}$. Since A is positive definite,

$$a_{ij} \le \sqrt{a_{ii}a_{jj}} < \sqrt{a_{jj}}.$$

The (i,i) element is overwritten by the result of evaluating the formula

$$x = c^2 a_{ii} - 2sca_{ij} + s^2 a_{jj},$$

or some simple rearrangement of it, where the computed c and s that are used have relative errors of order $\tilde{\gamma}_1$. Standard error analysis yields, for any order of evaluation,

$$\begin{aligned} |x - \widehat{x}| &\le \tilde{\gamma}_1(c^2|a_{ii}| + 2|sc||a_{ij}| + s^2|a_{jj}|) \\ &\le \tilde{\gamma}_1(|a_{ii}| + |a_{ij}| + |a_{jj}|) \\ &\le \tilde{\gamma}_1(1 + \sqrt{|a_{jj}|} + |a_{jj}|) \\ &\le \tilde{\gamma}_1(1 + \sqrt{n} + n) \\ &\le \tilde{\gamma}_n. \end{aligned}$$

Since $x = 1$ we conclude that replacing $\widehat{x}$ by 1 corresponds to a backward perturbation in A of order $\tilde{\gamma}_n$, which has no effect on the bounds in Lemma 4.1.

To summarize, Algorithm 3.1 has essentially perfect backward stability and we can explicitly set the diagonal elements to unity without affecting the stability.

Now we consider the implementation of the Bendel–Mickey algorithm in [2], [21]. Here, the Givens rotation in (3.1) is computed by

$$\begin{gathered} \alpha = a_{ij}^2 + \tfrac{1}{4}(a_{ii} - a_{jj})^2, \qquad \beta = \tfrac{1}{2}(a_{ii} + a_{jj} - 2)(a_{ii} - a_{jj}), \\ \gamma = \tfrac{1}{4}(a_{ii} + a_{jj} - 2)^2 - a_{ij}^2, \qquad \delta = \sqrt{\beta^2 - 4\alpha\gamma}, \\ \theta = \tfrac{1}{2}\cos^{-1}\left(\frac{\delta - \beta}{2\alpha}\right), \qquad c = \cos(\theta), \qquad s = \sin(\theta). \end{gathered}$$

It is clear that these formulae can involve serious cancellation and are not a stable way of computing θ. The computed Givens rotation will always be orthogonal to working precision, but when applied to A it will not necessarily accurately set a_{ii} to 1. In the Fortran implementation in [21] this weakness is recognized in so far as the routine checks how close the computed diagonal is to 1 and flags an error if a user-specified tolerance for the difference is exceeded.

We have experimented with the NAG library routine `G05GBF` (Mark 18) [24], which is based on the code from [21]. We called the code from MATLAB using a Mex interface and applied direct search optimization routines from the Test Matrix Toolbox [14]. We readily found examples where the computed diagonal is very far from 1. For example, in one run the vector of eigenvalues and the computed correlation matrix were

```
x =
    0.3844
    1.8365
    0.7791

A =
    1.0000   -0.2568   -0.6458
   -0.2568    0.9379    0.2772
   -0.6458    0.2772    1.0621
```

The eigenvalues of the matrix agree with the specified ones to the working precision ($u \approx 10^{-16}$), as expected, but the $(2,2)$ and $(3,3)$ elements are far from 1. In all such examples the NAG routine returns a nonzero error flag signalling the inaccurate diagonal.

According to the Guide to Available Mathematical Software (`http://gams.nist.gov`), the IMSL STAT/LIBRARY contains a routine `rncor` that generates a random correlation matrix with specified eigenvalues. The documentation provided on GAMS does not specify the method, but it says that an error flag indicates "Considerable loss of precision occurred in the rotations used to form the correlation matrix. Some of the diagonals of COR differ from 1.0 by more than the machine epsilon", which suggests that this routine is also based on the code from [21].

Our conclusion is that the Fortran code in [21] uses an unstable implementation of the Bendel–Mickey algorithm; this code and those based on it should be modified to obtain the Givens rotation from (3.4) and (3.5) and to explicitly set the diagonal to 1.

5 Generating (Cholesky) factors of correlation matrices.

Any correlation matrix $A \in \mathbb{R}^{n\times n}$ can be written X^TX, where $X \in \mathbb{R}^{m\times n}$ with $m \geq n$ has columns of unit 2-norm. Two-sided orthogonal transformations $A \leftarrow Q^TAQ$ correspond to one-sided transformations $X \leftarrow XQ$. Using this connection, we can derive an analogue of Algorithm 3.1 that constructs a random rectangular matrix with columns of unit 2-norm and specified singular values, provided that the squares of the singular values sum to n. This algorithm is of interest for at least two reasons. First, irrespective of rounding errors, the factor X always represents a positive semidefinite matrix $A = X^TX$ and its elements have half the dynamic range of those of A. For computational convenience, and to reduce storage, X can be taken to be square and upper triangular, that is, as the Cholesky factor of A (up to signs). Second, the one-sided Jacobi method for computing the singular value decomposition (SVD) has high accuracy properties analogous to those mentioned in Section 1 for the two-sided algorithm [7, 23]. In this case the relative error in the computed singular values satisfies a bound proportional to the condition number of the matrix scaled to have columns of unit 2-norm, and this matrix has 2-norm condition number within $\sqrt{n}$ of the smallest obtainable by column scaling. The factor X of a correlation matrix is therefore useful for testing the one-sided Jacobi method.

We denote the jth column of X by x_j.

Algorithm 5.1. *Given nonnegative scalars $\sigma_1, \dots, \sigma_n$ with $\sum_{i=1}^n \sigma_i^2 = n$, and an integer $m \geq n$, this algorithm produces a random matrix $X \in \mathbb{R}^{m\times n}$ having columns of unit 2-norm and singular values $\sigma_1, \dots, \sigma_n$. Optionally, X may be taken to be upper triangular.*

```
1. Form $X = U \operatorname{diag}(\sigma_i) V^T$, where $U \in \mathbb{R}^{m\times n}$ and $V \in \mathbb{R}^{n\times n}$ are random
   matrices with orthonormal columns from the Haar distribution.
2. while some $\|x_j\|_2 \neq 1$
         Find indices $i$ and $j$ with $i < j$ so that $\|x_i\|_2 < 1 < \|x_j\|_2$
         or $\|x_i\|_2 > 1 > \|x_j\|_2$.
         Form $a_{ii} = x_i^T x_i$, $a_{ij} = x_i^T x_j$, $a_{jj} = x_j^T x_j$.
         Construct a Givens rotation $Q = \left[\begin{smallmatrix} c & s \\ -s & c \end{smallmatrix}\right]$ in the $(i,j)$ plane
         according to (3.1), (3.4) and (3.5) to set $\|x_i\|_2 = 1$
         and transform $X \leftarrow XQ$.
   end
   Optionally, compute the QR factorization $X = \widetilde{Q}R$ and
   replace $X$ by $R(1\colon n, 1\colon n)$.
```

As for Algorithm 3.1, the while loop in Algorithm 5.1 is executed at most $n-1$ times. The cost of the algorithm is $m^3 + n^3$ flops for the initial formation of X (which is done without explicitly forming U and V—see [27], [15, Section 26.3]), plus at most $10n^2$ flops for the second stage (note that the column norms $\|x_j\|_2$ need only be computed once, at the start of the second stage, and can then be updated). The optional QR factorization costs a further $2n^2(m - n/3)$ flops (assuming the use of Householder transformations). Algorithm 5.1 is implemented in Matlab 6 as `gallery('randcolu',...)`.

Algorithm 5.1 has numerical stability and accuracy properties analogous to those of Algorithm 3.1.

We mention that Drmač [9, 8] constructs matrices of the form generated by Algorithm 5.1 by applying a one-sided, apparently infinite, product of Givens rotations to a diagonal matrix of singular values, but he gives no details of how this is done.

Acknowledgements.

We thank Sven Hammarling and Françoise Tisseur for comments on draft versions of this paper.

REFERENCES

1. R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 1994.
2. R. B. Bendel and M. R. Mickey, *Population correlation matrices for sampling experiments*, Comm. Statist. Simulation Comput., B7:2 (1978), pp. 163–182.
3. N. N. Chan and K.-H. Li, *Algorithm 115: A FORTRAN subroutine for finding a real symmetric matrix with prescribed diagonal elements and eigenvalues*, Comput. J., 26:2 (1983), pp. 184–186.
4. N. N. Chan and K.-H. Li, *Diagonal elements and eigenvalues of a real symmetric matrix*, J. Math. Anal. Appl., 91 (1983), pp. 562–566.
5. J. W. Demmel, *On floating point errors in Cholesky*, Technical Report CS-89-87, Department of Computer Science, University of Tennessee, Knoxville, TN, October 1989.
6. J. W. Demmel and A. McKenney, *A test matrix generation suite*, Preprint MCS-P69-0389, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, March 1989.
7. J. W. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13:4 (1992), pp. 1204–1245.
8. Z. Drmač, *A tangent algorithm for computing the generalized singular value decomposition*, SIAM J. Numer. Anal., 35:5 (1998), pp. 1804–1832.
9. Z. Drmač, *Accurate computation of the product induced singular value decomposition with applications*, SIAM J. Numer. Anal., 35:5 (1998), pp. 1969–1994.
10. P. A. Fillmore, *On similarity and the diagonal of a matrix*, Amer. Math. Monthly, 76 (1969), pp. 167–169.
11. G. E. Forsythe and E. G. Straus, *On best conditioned matrices*, Proc. Amer. Math. Soc., 6 (1955), pp. 340–345.
12. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
13. A. Greenbaum, *Iterative Methods for Solving Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.
14. N. J. Higham, *The Test Matrix Toolbox for* MATLAB *(version 3.0)*, Numerical Analysis Report No. 276, Manchester Centre for Computational Mathematics, Manchester, England, September 1995.
15. N. J. Higham, *Accuracy and Stability of Numerical Algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.
16. R. B. Holmes, *On random correlation matrices*, SIAM J. Matrix Anal. Appl., 12:2 (1991), pp. 239–272.
17. R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.
18. R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991.
19. K. D. Ikramov, *A remark on "A note on constructing a symmetric matrix with specified diagonal entries and eigenvalues"*, BIT, 38:4 (1998), p. 807.
20. R. S. Leite, T. R. W. Richa, and C. Tomei, *Geometric proofs of some theorems of Schur–Horn type*, Linear Algebra Appl., 286:149–173, 1999.

21. S. P. Lin and R. B. Bendel, *Algorithm AS 213: Generation of population correlation matrices with specified eigenvalues*, Applied Statistics, 34:2 (1985), pp. 193–198.

22. G. Marsaglia and I. Olkin, *Generating correlation matrices*, SIAM J. Sci. Stat. Comput., 5:2 (1984), pp. 470–475.

23. R. Mathias, *Accurate eigensystem computations by Jacobi methods*, SIAM J. Matrix Anal. Appl., 16:3 (1995), pp. 977–1003.

24. *NAG Fortran Library Manual, Mark 18.* NAG Ltd., Oxford, 1997.

25. J. R. Schott. *Matrix Analysis for Statistics.* Wiley, New York, 1997.

26. *Solutions to problem E 2741: Similarity and the diagonal of a matrix*, Amer. Math. Monthly, 87 (1980), pp. 62–63.

27. G. W. Stewart, *The efficient generation of random orthogonal matrices with an application to condition estimators*, SIAM J. Numer. Anal., 17:3 (1980), pp. 403–409.

28. William F. Trench, *Numerical solution of the eigenvalue problem for Hermitian Toeplitz matrices*, SIAM J. Matrix Anal. Appl., 10;2 (1989), pp. 135–146.

29. A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.

30. J. H. Wilkinson, *A priori error analysis of algebraic processes*, in Proc. International Congress of Mathematicians, Moscow 1966, I. G. Petrovsky, ed., Mir Publishers, Moscow, 1968, pp. 629–640.

31. H. Zha and Z. Zhang, *A note on constructing a symmetric matrix with specified diagonal entries and eigenvalues*, BIT, 35 (1995), pp. 448–452.