SIAM J. MATRIX ANAL. APPL.
Vol. 28, No. 4, pp. 1173–1198

ACCURATE FACTORIZATION AND EIGENVALUE ALGORITHMS FOR SYMMETRIC DSTU AND TSC MATRICES*

MARÍA JOSÉ PELÁEZ† AND JULIO MORO†

Abstract. Two algorithms are presented which compute, with small componentwise relative error, a block LDL^T factorization of symmetric matrices belonging to two classes of matrices: diagonally scaled totally unimodular (DSTU) and total signed compound (TSC) matrices. This accuracy is achieved by taking advantage of some special properties of such structures in order to avoid subtractions throughout the factorization process. Once an accurate block LDL^T decomposition is available, it is proved that one can easily obtain an accurate symmetric rank-revealing decomposition, which is the starting point for algorithms computing with high relative accuracy the eigenvalues and eigenvectors of arbitrary, possibly indefinite, symmetric matrices. This proves that eigenvalues and eigenvectors of symmetric DSTU and TSC matrices can be computed with high relative accuracy.

Key words. symmetric eigenproblem, high relative accuracy, rank-revealing decomposition, structured matrices

AMS subject classifications. 65F15, 15A18, 15A23

DOI. 10.1137/050631537

1. Introduction.

It is well known that, given an arbitrary real square matrix, conventional, general-purpose eigenvalue algorithms like QR or divide-and-conquer can only guarantee full accuracy in the computation of the eigenvalues with largest absolute value. If one is interested in obtaining *all* eigenvalues with correct sign and correct leading digits, then more specifically devised algorithms are needed. Given a matrix $A \in \mathbb{R}^{n\times n}$, the best one can expect is that all computed eigenvalues $\widehat{\lambda}_i$ and corresponding eigenvectors $\widehat{q}_i$, $i = 1, \ldots, n$, satisfy

$$\begin{aligned} \frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} &= O(\kappa\epsilon), \\ \Theta(\widehat{q}_i, q_i) &= \frac{O(\kappa\epsilon)}{\operatorname{relgap}(\lambda_i)}, \end{aligned} \tag{1}$$

where λ_i, q_i and $i = 1, \ldots, n$, are, respectively, the exact eigenvalues and eigenvectors of A; ϵ is the machine precision; $O(\cdot)$ is the customary big-oh Landau notation; $\Theta(\cdot, \cdot)$ stands for the angle between two n-vectors; $\operatorname{relgap}(\lambda_i)$ is the usual relative gap $\operatorname{relgap}(\lambda_i) = \min\{\min_{j\neq i} \frac{|\lambda_j - \lambda_i|}{|\lambda_i|}, 1\}$, $i = 1, \ldots, n$; and κ is some constant of moderate size, eventually depending on the dimension n but independent of the condition number of A. The algorithms achieving these, or some slightly relaxed, error bounds are the so-called *high relative accuracy algorithms* for the eigenvalue problem. A similar definition can be given for high relative accuracy algorithms for the singular value decomposition (SVD), replacing eigenvalues by singular values.

Several high relative accuracy algorithms are known so far, both for the SVD and for the standard eigenvalue problem. Their scope, however, is limited, in the

*Received by the editors May 17, 2005; accepted for publication (in revised form) by J. Barlow October 9, 2006; published electronically December 18, 2006. The research conducted in this paper was supported by Spanish Ministerio de Ciencia y Tecnología grant BFM-2003-00223.
http://www.siam.org/journals/simax/28-4/63153.html
†Departamento de Matemáticas, Universidad Carlos III de Madrid, 28911 Leganés, Spain (mpelaez@math.uc3m.es, jmoro@math.uc3m.es).

sense that each algorithm is specifically designed for a particular class of matrices, taking advantage of the structure properties specific to that class in order to preserve the accuracy of the computation. Some such classes for the SVD are, for instance, bidiagonal matrices [8, 17], matrices of the form BD with D diagonal and B well-conditioned [10, 16, 23], acyclic, Cauchy and Vandermonde matrices [11], unit-displacement-rank matrices [7], or weakly diagonally dominant M-matrices [9, 24]. Some of these classes can be grouped into a larger class of matrices for which it is possible to compute accurately a rank-revealing decomposition [11]. For the eigenvalue problem, some classes allowing accurate eigendecomposition are symmetric scaled diagonally dominant [4]; symmetric positive definite [10]; symmetric tridiagonal [21, 12, 13]; symmetric Cauchy and Vandermonde [15]; matrices of the form $H = BD$, where D is diagonal and B is well conditioned [30]; and symmetric indefinite matrices allowing for an accurate initial factorization [25, 27, 14]. So far, the only class of nonsymmetric matrices whose eigenvalues can be computed to high relative accuracy is the class of totally nonnegative matrices [22].

Among all these different methods we will concentrate on a family of algorithms [11, 14, 25] which proceed in two stages: In the first stage, the algorithm computes an initial factorization of the matrix. Then an appropriately chosen Jacobi-type algorithm is applied to the factors. To achieve the accuracy bounds (1), both stages must be performed accurately enough. The accuracy of the second stage is usually ensured once and for all through a detailed error analysis, valid for any factorized matrix. Once this is done, the accuracy of the overall algorithm depends entirely on the accuracy of the preprocessing factorization in the first stage. In other words, *the classes of matrices such that these algorithms compute all its eigenvalues and eigenvectors to high relative accuracy become those classes for which it is known how to compute a sufficiently accurate initial factorization.*

In the case of the SVD, for instance, several classes of matrices were identified in [11] such that special versions of Gaussian elimination with complete pivoting (GECP), conveniently adapted to each class, lead to accurate *nonsymmetric* factorizations of the form $A = X\Delta Y^T$. Two of these classes are the *diagonally scaled totally unimodular* (DSTU) matrices and *total signed compound* (TSC) matrices (see sections 3.1 and 3.2 below for definitions). Our main contribution in this paper is to prove that, for both DSTU and TSC structures, the subclass of *symmetric* matrices allows for the computation of *symmetric* factorizations of the form $A = X\Delta X^T$ in a sufficiently accurate way to compute eigenvalues and eigenvectors with the accuracy (1). Therefore, for any symmetric matrix which is either DSTU or TSC, all its eigenvalues and eigenvectors can be computed to high relative accuracy. The leading idea of the factorization algorithms we propose is, as in [11], to take advantage of the special properties of each structure in order to completely avoid subtraction throughout the factorization process.

Two high relative accuracy algorithms are available to compute eigenvalues and eigenvectors of general, possibly indefinite, symmetric matrices: the *J-orthogonal algorithm* [29, 25] and the *signed SVD algorithm* [14]. None of them, strictly speaking, has error bounds of the form (1). For the J-orthogonal method, the constant κ in (1) is the maximum of the condition numbers of some intermediate matrices produced by the algorithm, and these condition numbers could be, in principle, arbitrarily large. The signed SVD method, on the other hand, has an error bound for eigenvectors of the form (1), but with a potentially smaller quantity, relgap*, in the denominator instead of the usual relative gap (see (9) below). Nevertheless, both algorithms are

able in practice to compute eigenvalues and eigenvectors to high relative accuracy.

Both algorithms begin by initially factorizing the matrix, although in a slightly different way. To be more precise, we begin by defining *rank-revealing decompositions.*

DEFINITION 1. *Given $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, a rank-revealing decomposition (RRD) of A is any factorization $A = X\Delta Y^T$ such that $X \in \mathbb{R}^{m\times r}$, $Y \in \mathbb{R}^{n\times r}$, $\Delta \in \mathbb{R}^{r\times r}$ for $r \leq \min\{m, n\}$, where Δ is diagonal and nonsingular and both matrices X, Y are well-conditioned.*

For instance, the SVD is an RRD factorization, but others can be obtained, for instance, via Gaussian elimination with complete pivoting (GECP), QR with complete pivoting, or, as we will see, via the diagonal pivoting method.

The signed SVD method begins by computing an RRD, either symmetric with $X = Y$, or nonsymmetric with $X \neq Y$.[1] Since in our case the matrix A is symmetric, preservation of structure advises keeping the symmetry in the factorization. Therefore, we restrict ourselves in this paper to the analysis of *symmetric* RRDs of the form

$$A = X\Delta X^T. \tag{2}$$

The J-orthogonal algorithm, on the other hand, begins by computing a so-called *symmetric indefinite factorization*

$$PAP^T = GJG^T, \tag{3}$$

where P is a permutation matrix, J is square diagonal with diagonal elements ± 1 and G has full column rank. Although this is not exactly an RRD, its computation is equivalent to computing the symmetric RRD above, since it suffices to scale Δ on both sides with $|\Delta|^{-1/2} = \mathrm{diag}(|\Delta_{ii}|^{-1/2})$ to obtain $A = GJG^T$, where $G = X|\Delta|^{1/2}$ and $J = |\Delta|^{-1/2}\Delta|\Delta|^{-1/2}$. Therefore, we concentrate in what follows in obtaining a symmetric RRD (2).

The way we obtain the symmetric RRD is via the so-called *block LDL^T factorization*

$$PAP^T = LDL^T, \tag{4}$$

where P is a permutation matrix, L is unit lower triangular, and D is block-diagonal with 1×1 and 2×2 diagonal blocks. Of course, this is not an RRD, since D is not diagonal. However, it suffices to orthogonally diagonalize the 2×2 blocks in D via Givens rotations to obtain an RRD, a procedure which was introduced in [26, 27] to obtain symmetric indefinite decompositions: let $Q \in \mathbb{R}^{n\times n}$ be an orthogonal, block-diagonal matrix conformal to D, each 2×2 diagonal block of Q being the Givens rotation used to diagonalize the corresponding diagonal block of D. Then $A = X\Delta X^T$ with

$$X = P^T LQ \qquad \text{and} \qquad \Delta = Q^T DQ. \tag{5}$$

Notice that L and X have the same condition number in any unitarily invariant norm.

[1] There might be advantages in computing a nonsymmetric RRD of a symmetric matrix, due to the additional freedom in pivoting: a class of structured symmetric matrices might allow for accurate *nonsymmetric* RRDs but not for accurate symmetric ones. Whether such a class exists, however, is still an open question.

One can easily show,[2] adapting the proof of [11, Theorem 2.1] from the SVD context to the eigenvalue problem, that the symmetric RRD (2) determines the eigendecomposition to high relative accuracy, i.e., that, as stated in [11] for the SVD, having any symmetric RRD is as good as having an eigendecomposition, because any small change (in the sense given by (7) below) in the factors of the RRD produces small changes in the eigenvalues and eigenvectors.

Once we have this, the error analyses of both the J-orthogonal and the signed SVD method guarantee the accuracy of the computed eigenvalues and eigenvectors *only if* the initial factorization is computed accurately enough. Proving this for DSTU ad TSC matrices is our goal in the present paper, but since the error analyses of both methods are very different, the accuracy requirements on the factorizations are also diverse. We will deal in what follows only with the signed SVD method, whose error analysis is more amenable to our approach, but we stress that similar results hold for the J-orthogonal method.

To analyze the accuracy of the computed RRD we proceed in two stages: First, we see how to compute block LDL^T factorizations of symmetric DSTU and TSC matrices with *componentwise* small relative error, i.e., if $\widehat{L}$ and $\widehat{D}$ are the factors computed in floating point arithmetic and L and D are the exact factors, it will be shown that

$$|\widehat{l}_{ij} - l_{ij}| = O(\epsilon)|l_{ij}|, \qquad |\widehat{d}_{ij} - d_{ij}| = O(\epsilon)|d_{ij}| \tag{6}$$

for every $i, j \in \{1, \ldots, n\}$. To prove this it is enough to show that *no subtraction is ever performed throughout the factorization process.* Products, quotients, square roots, and sums of quantities of like sign are harmless operations from the point of view of producing large forward errors. The only possible source of forward instability is cancellation, and we will rule it out by avoiding subtraction (even if the subtracted quantities are not close to each other).

In a second stage, we will show that the RRD obtained from the block LDL^T factorization via Givens diagonalization, as in (5), satisfies the requirements which ensure the accuracy (1). According to the error analysis in [14], these requirements are that the factor Δ in (2) be computed with small *componentwise* relative errors, and the factor X be computed with small *normwise* relative error in any norm, i.e.,

$$\|\widehat{X} - X\| = O(\epsilon)\|X\|, \qquad |\widehat{\Delta}_{ii} - \Delta_{ii}| = O(\epsilon)|\Delta_{ii}|, \quad i = 1, \ldots, n, \tag{7}$$

if $\widehat{X}, \widehat{\Delta}$ are the factors computed in floating point arithmetic and X, Δ are the exact ones (actually, we will prove a sharper bound for X in Theorem 6). All this leads to the conclusion that *all eigenvalues and eigenvectors of symmetric DSTU and TSC matrices can be computed with high relative accuracy via the signed SVD method, provided the initial RRD is computed with these special factorization algorithms.* To be more precise, the eigenvalues are computed with an error of the form (1), where κ is given by

$$\kappa = \kappa(R')\kappa(X), \tag{8}$$

[2]This has been done in [15]: if $A = XDX^T$ and $\widetilde{A} = \widetilde{X}\widetilde{D}\widetilde{X}^T$ are RRDs of the symmetric matrices A and $\widetilde{A}$, and both the normwise relative error $\|\widetilde{X} - X\|/\|X\|$ in X and the componentwise relative error $|\widetilde{D}_{ii} - D_{ii}|/|D_{ii}|$ in D are bounded by a quantity β smaller than 1, then, setting $\eta = \beta(2 + \beta)\kappa(X)$, the relative error in the eigenvalues is bounded by $O(\eta)$ and the sine of the canonical angles between the eigenvectors of A and $\widetilde{A}$ is bounded by $O(\eta)$ divided by the relative gap (see [15, section 2] for more details).

$\kappa(\cdot)$ denotes the condition number in the two-norm, X is the nondiagonal factor in (2), and R' is the best conditioned row diagonal scaling of the triangular factor R of a QR factorization with column pivoting of the product $X\Delta$. Since it was proved in [11, Theorem 3.2] that $\kappa(R')$ is at most of order $O(n^{3/2}\kappa(X))$, we see that, up to a moderate constant, the factor κ is of the order of the condition number of the factor X in the RRD. Therefore, it is important to guarantee that $\kappa(X)$ is moderate. We will do this for symmetric DSTU matrices in Theorem 3, proving that $\kappa(X) = O(n^2)$. No such bound is available so far for TSC matrices.

As for the eigenvectors, they are computed with an error of the form (1) but replacing the usual relative gap with

$$\text{relgap}^*(|\lambda_i|) = \min\left\{ \min_{\substack{j\in\mathcal{S}\\ j\neq i}} \left| \frac{|\lambda_j| - |\lambda_i|}{\lambda_i} \right|, 1 \right\}, \tag{9}$$

where the index set $\mathcal{S}$ is equal to $\{1,\ldots,n\}$ unless the eigenvalue, say λ_{j_0}, whose absolute value is closest to $|\lambda_i|$ has opposite sign to λ_i. In that case, $\mathcal{S}$ is obtained from $\{1,\ldots,n\}$ by removing j_0 and the index k of any other eigenvalue within a relative distance of order $O(\kappa\epsilon)$ of λ_{j_0}.

The paper is organized as follows. We begin in section 2 with a brief review of some basic properties of the block LDL^T factorization, which will be needed later on. Then, we show in section 3 how to achieve accurate block LDL^T decompositions of symmetric DSTU and TSC matrices. The factorization algorithm for $n \times n$ DSTU matrices has a computational cost of order $O(n^3)$, while the one for TSC matrices has a worst-case cost of $O(n^4)$. Section 4 is devoted to showing that, given any block LDL^T factorization satisfying (6), all further manipulations required to derive (2) from (4) do not spoil the accuracy. More precisely, we show that the componentwise accuracy is preserved in Δ, and it is tranformed at worse in columnwise accuracy for X. Some numerical experiments are presented in section 5 which confirm the high accuracy of the proposed algorithms. Finally, we collect in Appendix A the proofs of the results in section 2.

We end this introduction with a brief comment on singular matrices: we will assume that all matrices A under examination are nonsingular. If A is singular, the number of zero eigenvalues is determined from any RRD satisfying (7), and the signed SVD method can be enhanced to compute the null vectors using a complete QR factorization with complete pivoting. However, this is out of the scope of our error analysis in this paper.

2. Block LDL^T factorizations of symmetric matrices. One of the possible symmetric analogues of the LU decomposition is the block LDL^T decomposition (4). Any symmetric matrix admits such a factorization [18, Chapter 11], and the most common procedure to compute it is the *diagonal pivoting method*: it begins by choosing a permutation matrix P, an integer $s = 1$ or $s = 2$, and an $s \times s$ nonsingular pivot matrix E such that

$$PAP^T = \begin{pmatrix} E & C^T \\ C & B \end{pmatrix},$$

so

$$PAP^T = \begin{pmatrix} I_s & 0 \\ CE^{-1} & I_{n-s} \end{pmatrix} \begin{pmatrix} E & 0 \\ 0 & B - CE^{-1}C^T \end{pmatrix} \begin{pmatrix} I_s & E^{-1}C^T \\ 0 & I_{n-s} \end{pmatrix}. \tag{10}$$

The block LDL^T factorization of A follows from simply repeating this same process on the succesive $(n-s)\times(n-s)$ Schur complements $B-CE^{-1}C^T$. The whole process costs $n^3/3$ arithmetic operations plus the cost of determining the permutations.

Several symmetric strategies are available for choosing the pivot matrix E, analogous to either partial, complete, or rook pivoting in LU. Since our final goal is an RRD, we will mostly use the Bunch–Parlett pivoting strategy [3], a symmetric analogue of complete pivoting which usually produces well-conditioned factors L (see [2] for a detailed analysis of element growth of the L factor). This pivoting strategy can be summarized as follows:

$$
\begin{array}{ll}
 & \alpha = (1+\sqrt{17})/8 \ 0.64 \\
 & \mu_0 = \max_{i,j}|a_{ij}| =: |a_{pq}| \\
 & \mu_1 = \max_i |a_{ii}| =: |a_{rr}| \\
(11) & \textbf{If } \mu_1 \geq \alpha\mu_0 \textbf{ then} \\
 & \qquad \text{choose } E = [a_{rr}] \text{ as } 1\times 1 \text{ pivot} \\
 & \textbf{else} \\
 & \qquad \text{choose } E = \begin{pmatrix} a_{pp} & a_{pq} \\ a_{pq} & a_{qq} \end{pmatrix} \text{ as } 2\times 2 \text{ pivot}
\end{array}
$$

Any 2×2 pivot E chosen with this strategy is a symmetric indefinite, well-conditioned matrix whose condition number is bounded by $(1+\alpha)/(1-\alpha)\approx 4.6$ in the 2-norm. The value $(1+\sqrt{17})/8$ of the constant α is chosen to ensure that the growth factor corresponding to two successive 1×1 pivots equals the growth factor corresponding to a 2×2 pivot.

It is well known that any final or intermediate value computed by Gaussian elimination with any pivoting strategy is either a minor or a quotient of minors of the original matrix (see Lemma 5.1 in [11]). This is a consequence of the properties of Schur complements: given any matrix A (eventually nonsymmetric), partitioned as

$$
A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},
$$

with A_{11} square and nonsingular, the *Schur complement*[3] of A_{11} in A is

$$
\mathcal{C} = A_{22} - A_{21}A_{11}^{-1}A_{12}. \tag{12}
$$

One of the simplest properties of $\mathcal{C}$ is that $\det A = \det A_{11}\det\mathcal{C}$ or, equivalently, that $\det\mathcal{C} = \det A/\det A_{11}$. Hence, $\det\mathcal{C}$ is a quotient of minors of A. To prove that every intermediate quantity in the block LDL^T factorization (4) is also a quotient of minors of the original matrix, some properties of Schur complements are needed. Due to their technical character, all proofs are postponed to Appendix A. We will use MATLAB notation to state the results; i.e., $A([r_1,\dots,r_p],[c_1,\dots,c_p])$ denotes the $p\times p$ submatrix of A containing the elements in rows $r_1,\dots,r_p$ and columns $c_1,\dots,c_p$. We also abbreviate as $1:k$ the list of all integers from 1 to k.

LEMMA 1. *Let $A\in\mathbb{R}^{n\times n}$, let $k<n$ and let A_k be the upper left $k\times k$ principal submatrix of A. Then*

[3]In order not to complicate the notation, the definition (12) and Lemma 1 are given in terms of Schur complements of the upper left leading principal submatrix, which is the only one we will employ below. Of course, all results hold true if A_{11} is replaced by any square nonsingular submatrix of A.

(a) *the* (i,j) *element of the* $(n-k)\times(n-k)$ *Schur complement* $\mathcal{C}_k$ *of* A_k *in* A *is given by*

$$\mathcal{C}_k(i,j) = \frac{\det A([1:k,k+i],[1:k,k+j])}{\det A([1:k],[1:k])} \tag{13}$$

for each $i,j \in \{1,\dots,n-k\}$*;*

(b) *for any* $s \le n-k$*, the minor of* $\mathcal{C}_k$ *containing rows* $i_1 < \cdots < i_s$ *and columns* $j_1 < \cdots < j_s$ *is given by*

$$\det \mathcal{C}_k([i_1,\dots,i_s],[j_1,\dots,j_s]). \tag{14}$$
$$= \frac{\det A([1:k,k+i_1,\dots,k+i_s],[1:k,k+j_1,\dots,k+j_s])}{\det A[1:k],[1:k]}.$$

In particular, any minor of $\mathcal{C}_k$ *is a quotient of minors of* A*.*

As a consequence of Lemma 1, we prove our previous claim.

THEOREM 1. *Let* A *be a real symmetric matrix and let* $PAP^T = LDL^T$ *be a block* LDL^T *factorization of* A *as described in* (4)*, obtained using any pivoting strategy. Then every entry of* L *or* D *is either zero or a quotient of minors (or just a minor) of* A*.*

It is interesting to observe at this point that the block LDL^T factorization does not enjoy all the good properties of the LDU decomposition obtained from Gaussian elimination. For instance, it is not true, as it is for Gaussian elimination, that every minor of L is a quotient of minors of A: if we consider the symmetric matrix

$$A = \begin{pmatrix} 13 & 39 & 65 \\ 39 & 128 & 274 \\ 65 & 274 & 903 \end{pmatrix}$$

which can be factorized as $A = LDL^T$ with

$$L = \begin{pmatrix} 1 & & \\ 3 & 1 & \\ 5 & 7 & 1 \end{pmatrix}, \qquad D = \left(\begin{array}{c|cc} 13 & & \\ \hline & 11 & 2 \\ & 2 & 11 \end{array}\right),$$

one can check that

$$\det L([2,3],[1,2]) = \begin{vmatrix} 3 & 1 \\ 5 & 7 \end{vmatrix} = 16$$

is not a quotient of minors of A.

We finish this section with explicit formulas for the elements of L and D as quotients of minors of A. These formulas, which will be needed later to recompute the entries of L, are, to our knowledge, new in the literature. They are an extension of classical formulas for Gaussian elimination (see, e.g., [20, section 1.4]). Such classical formulas no longer hold in our case, since, as shown by the example above, not every minor of L is a quotient of minors of A, and this is an essential ingredient of the proofs for Gaussian elimination. For the sake of simplicity, the formulas are written with no reference to the pivoting permutations, but the same result holds for the factorization (4), appropriately renaming rows and columns according to the permutation matrix P.

LEMMA 2. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix factorized as $A = LDL^T$ with $L \in \mathbb{R}^{n\times n}$ unit lower triangular and $D \in \mathbb{R}^{n\times n}$ block-diagonal with 1×1 and 2×2 diagonal blocks. Let $D = \operatorname{diag}(D_1, \ldots, D_r)$ with $D_k \in \mathbb{R}^{s_k\times s_k}$, $s_k = 1$ or 2, $k = 1, \ldots, r$, and partition L conformally as $L = [L_1 \,|\, \cdots \,|\, L_r]$ with $L_k \in \mathbb{R}^{n\times s_k}$. For each $k \in \{1, \ldots, r\}$, set $n_k = s_1 + \ldots + s_k$. Then, for each $k \in \{1, \ldots, r\}$, the elements (i, j) of L with $j \in \{n_k - s_k + 1, \ldots, n_k\}$ are given by*

$$L(i,j) = \frac{\det A([1:j-1, i, j+1:n_k], [1:n_k])}{\det A([1:n_k], [1:n_k])}, \qquad i = n_k + 1, \ldots, n, \tag{15}$$

and if $n_0 = 0$, the elements (i, j) of D with $i, j \in \{n_k - s_k + 1, \ldots, n_k\}$ are given by

$$D(i,j) = \frac{\det A([1:n_{k-1}, i], [1:n_{k-1}, j])}{\det A([1:n_{k-1}], [1:n_{k-1}])}. \tag{16}$$

3. Accurate factorization of symmetric DSTU and TSC matrices. In this section we show that, for both DSTU and TSC matrices, the block LDL^T factorization (4) can be computed with *componentwise* small relative error, as stated in (6). To prove it, we will modify the diagonal pivoting method in such a way that no subtraction is ever performed throughout the factorization process. Since cancellation is the only possible source of forward instability, the overall process will produce a small relative forward error in each single component of L and D.

3.1. Diagonally scaled totally unimodular matrices.

DEFINITION 2. *A matrix Z with integer entries is called* totally unimodular (TU) *if all its minors are -1, 0, or 1. In particular, the entries of Z must be either -1, 0, or 1. A matrix A is* diagonally scaled totally unimodular (DSTU) *if $A = \mathcal{D}_L Z \mathcal{D}_R$, where Z is totally unimodular, and $\mathcal{D}_L$ and $\mathcal{D}_R$ are diagonal matrices.*

The class of TU matrices contains some well-known classes as particular cases (see, e.g., [1, section 2.3]). Some of them are acyclic matrices, finite element matrices from linear mass-spring systems, or reduced node-arc incidence (RNAI) matrices [28]. In our case we are interested only in *symmetric* DSTU matrices, i.e., symmetric matrices A which can be written as

$$A = \mathcal{D} Z \mathcal{D}, \tag{17}$$

where Z is symmetric TU and $\mathcal{D} = \operatorname{diag}(d_1, \ldots, d_n)$ is diagonal. The matrix Z is supposed to be known exactly, but the elements of $\mathcal{D}$ are known only to high relative accuracy.

A first important property of DSTU matrices is that, as can be easily checked, *any Schur complement of a DSTU is still DSTU*. Hence, we may use the special properties of DSTU matrices at any stage of the factorization. The second important property, which is the key to avoid subtraction, is the following.

LEMMA 3. *Any minor of a symmetric DSTU matrix* (17) *is a monomial with coefficients $0, 1$ or -1 in the diagonal elements d_i of $\mathcal{D}$.*

As a consequence of this, any minor of a DSTU matrix is determined to high relative accuracy, and so are the elements of any of its Schur complements since, according to Theorem 1, each of them is just a quotient of minors of the original matrix.

Another interesting property of DSTU matrices is that the 2×2 pivots chosen by the Bunch–Parlett strategy have a very special structure.

LEMMA 4. *Any* 2×2 *pivot choosen by the Bunch–Parlett pivoting strategy on a symmetric DSTU matrix has, at least, one zero entry on the diagonal.*

Proof. A 2×2 pivot is chosen whenever

$$\mu_1 = \max_i |a_{ii}| := |a_{rr}| < \alpha\, |a_{pq}| =: \alpha \max_{i,j} |a_{ij}| = \alpha\, \mu_0.$$

If we suppose that both a_{pp} and a_{qq} are nonzero, then

$$|a_{pp}| \le \mu_1 < \alpha\, |a_{pq}|, \qquad |a_{qq}| \le \mu_1 < \alpha\, |a_{pq}|, \tag{18}$$

so $a_{pq} = d_p d_q z_{pq}$ is nonzero and, consequently, also d_p and d_q are different from zero. This, together with (18), implies that

$$|d_p| < \alpha^2 |d_p|, \tag{19}$$

in contradiction with the fact that $\alpha < 1$. Hence, either $a_{pp} = 0$ or $a_{qq} = 0$. ∎

Notice that inequality (19) is also in contradiction with $\alpha = 1$. Therefore, *Lemma 4 holds even if* $\alpha = 1$, a fact we will need in section 3.1.2 once we slightly modify the pivoting strategy.

3.1.1. Accurate block LDL^T factorization of symmetric DSTU matrices. We distinguish two cases, depending on the size of the pivot chosen at each stage.

- *Case 1: The chosen pivot* $E = [a_{rr}]$ *is* 1×1.
  Notice first that the elements of L computed at this stage are just a quotient
  $$l_{ir} = (CE^{-1})_{ir} = \frac{a_{ir}}{a_{rr}}$$
  of elements of the Schur complement computed in the previous stage. Therefore, l_{ir} is computed with small forward error, provided a_{ir} and a_{rr} are computed with small forward error as well. Computing the elements of D, however, may involve subtraction, since they are given by the formulas
  $$(B - CE^{-1}C^T)_{ij} = a_{ij} - l_{ir} a_{rj}. \tag{20}$$
  According to Theorem 1 and Lemma 3, each of the operands above is either zero or, up to a sign, a product of powers of the diagonal elements of $\mathcal{D}$. Actually, a simple computation shows that each operand is a monomial with coefficients ± 1 or 0 in the two variables d_i and d_j. Hence, (20) can be rewritten as
  $$m_1 = m_2 + m_3$$
  where each operand m_i for $i = 1, 2, 3$ is a monomial with coefficient ± 1 or 0 in the two variables d_i and d_j. There are four possibilities for this arithmetic operation, depending on whether m_2 and m_3 are zero or not. Three of the possibilities give rise to no operation at all. The fourth possibility, in which both m_2 and m_3 are nonzero, can have only $m_1 = 0$ as the result of the arithmetic operation, since the only way to obtain a coefficient $1, -1$, or 0 in the monomial m_1 is that the nonzero coefficients of m_2 and m_3 are ± 1 and cancel each other. In other words, *whenever the arithmetic operation* (20) *has two nonzero operands we assign the result to zero without performing the arithmetic operation.* Thus we avoid the possible cancellation which this operation might have produced.

- *Case* 2*: The chosen pivot* E *is* 2×2 *with largest off-diagonal element* a_{pq}*,* $p < q$*.*
  Taking into account Lemma 4, the entries in the two columns of L are computed as

$$l_{ip} = (CE^{-1})_{ip} = \frac{-a_{ip}a_{qq}}{a_{pq}^2} + \frac{a_{iq}a_{pq}}{a_{pq}^2}, \tag{21}$$

$$l_{iq} = (CE^{-1})_{iq} = \frac{a_{ip}a_{pq}}{a_{pq}^2} - \frac{a_{iq}a_{pp}}{a_{pq}^2}. \tag{22}$$

  Again, both expressions can be written in the form $m_1 = m_2 + m_3$, where each m_i is a monomial with coefficients $0, 1$, or -1 in three variables, namely d_i and the reciprocals of d_p and d_q. The same argument employed above applies here, i.e., we can avoid any potential subtraction by setting $m_1 = 0$ whenever both operands m_2 and m_3 are nonzero.
  Something similar happens with the elements of D: the elements of the Schur complement of an arbitary matrix are of the form

$$(B - CE^{-1}C^T)_{ij}$$

$$= a_{ij} - \frac{a_{ip}a_{qq}a_{pj} - a_{iq}a_{pq}a_{pj} - a_{ip}a_{pq}a_{qj} + a_{iq}a_{pp}a_{qj}}{a_{pp}a_{qq} - a_{pq}^2}, \tag{23}$$

  but in our case, due to Lemma 4, we have $a_{pp}a_{qq} = 0$. Replacing this fact in (23), we obtain that the entries of the Schur complement are a sum

$$m_1 = m_2 + m_3 + m_4 + m_5 \tag{24}$$

  of at most four operands, each of which is a monomial in the variable $d_i d_j$ with coefficients ± 1 or 0, as is the result m_1 of (24). If all four operands are zero, or if there is a single nonzero operand, no operation is performed. If we have exactly two or four nonzero operands in (24), the same argument employed above implies that m_1 must be zero, since the only possible sum in $\{1, -1, 0\}$ of two or four numbers in $\{1, -1\}$ is zero. Finally, in the case when exactly three operands are nonzero, two of them must necessarily cancel each other, and the result m_1 is equal to the third operand. Therefore, if the three nonzero operands are m_r, m_s, m_t, then we can assign

$$m_1 = -|m_t|\text{sign}(m_r m_s m_t),$$

  where m_t is any of the three operands, since all three have the same absolute value $|d_i d_j|$.

Thus we have proved the following result.

THEOREM 2. *Algorithm* 1 *computes all entries of the factors* L *and* D *of the block* LDL^T *factorization of a symmetric DSTU matrix to high relative accuracy, i.e.,*

$$|\widehat{l}_{ij} - l_{ij}| = O(\epsilon)|l_{ij}|, \quad |\widehat{d}_{ij} - d_{ij}| = O(\epsilon)|d_{ij}|,$$

where $\widehat{L}$ *and* $\widehat{D}$ *are the factors computed in floating point arithmetic by Algorithm* 1 *and* L*,* D *are the the exact factors which the diagonal pivoting method would compute*

in exact arithmetic choosing the pivots with the same dimensions and positions as those chosen in floating point arithmetic to compute $\widehat{L}$ *and* $\widehat{D}$.

It should be noted that any attempt to theoretically estimate the constants inside the $O(\epsilon)$ in Theorem 2 is bound to be pessimistic. Take, for instance, the number p_k of floating point operations required to compute the elements of D at stage k of the LDL^T factorization. This quantity is given by the recursive formula $p_{k+1} = 2p_k + 1$, so $p_k = 2^{k+1} - 1$. Therefore, the constant inside the $O(\epsilon)$ given by a straightforward error analysis would be exponential. However, this is never observed in practice.

Finally, the previous analysis suggests the following $O(n^3)$ algorithm.

ALGORITHM 1. BLOCK LDL^T FACTORIZATION OF A SYMMETRIC DSTU MATRIX A.

Input: symmetric $n \times n$ DSTU matrix A

Output: unit lower triangular matrix L, block diagonal matrix D with 1×1 and 2×2 diagonal blocks, permutation matrix P such that $PAP^T = LDL^T$.

for $i = 1$ **to** n
choose pivot according to Bunch–Parlett pivoting strategy
  if 1×1 pivot a_{ii}
    $D_{ii} = a_{ii}$
    for $j = i + 1$ **to** n
      $l_{ji} = a_{ji}/a_{ii}$
    endfor
    for $j = i + 1$ **to** n
      for $k = i + 1$ **to** n
        $a_{jk} = a_{jk} - \dfrac{a_{ji}a_{ik}}{D_{ii}}$
        (*)If the last subtraction has two nonzero operands, set $a_{jk} = 0$
      endfor
    endfor
  elseif 2×2 pivot, $\begin{pmatrix} a_{ii} & a_{i,i+1} \\ a_{i+1,i} & a_{i+1,i+1} \end{pmatrix}$
    $D_{ii} = a_{ii}, D_{i,i+1} = D_{i+1,i} = a_{i,i+1}, D_{i+1,i+1} = a_{i+1,i+1}$
    for $j = i + 1$ **to** n
      $l_{ji} = \dfrac{a_{j,i+1}a_{i,i+1}}{a_{i,i+1}^2} - \dfrac{a_{ji}a_{i+1,i+1}}{a_{i,i+1}^2}$
      (*) If the last subtraction has two nonzero operands, set $l_{ji} = 0$
    endfor
    for $j = i + 2$ **to** n
      $l_{j,i+1} = \dfrac{a_{j,i}a_{i,i+1}}{a_{i,i+1}^2} - \dfrac{a_{j,i+1}a_{ii}}{a_{i,i+1}^2}$
      (*) If the last subtraction has two nonzero operands, set $l_{j,i+1} = 0$
    endfor
    for $j = i + 1$ **to** n
      for $k = i + 1$ **to** n
        $m_2 = a_{jk}, m_3 = -\dfrac{a_{jq}a_{pq}a_{pk}}{a_{pq}^2}, m_4 = -\dfrac{a_{jp}a_{pq}a_{qk}}{a_{pq}^2}$
        if $a_{pp} = 0$
          $m_5 = \dfrac{a_{jp}a_{qq}a_{pk}}{a_{pq}^2}$

```
29.                    else m_5 = a_jq a_pp a_qk / a_pq^2
30.                endif
31.                a_jk = m_2 + m_3 + m_4 + m_5
32.                (*) If the last addition has two or four nonzero operands,
                       set a_jk = 0
33.                (*) If the last subtraction has three nonzero operands m_r, m_s, m_t,
34.                   set a_jk = -|m_r| sign(m_r m_s m_t)
35.             endfor
36.          endfor
37.       endif
38. endfor
```

3.1.2. A new pivoting strategy. In addition to its accuracy, another feature of the GECP decomposition $PAP^T = LDU$ of nonsymmetric $n \times n$ DSTU matrices is that the condition numbers of L and U grow at most quadratically with the dimension n [11, Theorem 10.2]. To prove the same for the triangular factor L in the block LDL^T decomposition we will slightly change the pivoting strategy. Consider the following one:

$$\begin{array}{l} \textbf{if} \quad \mu_0 = \max_{i,j}|a_{ij}| = \max_i |a_{ii}| = \mu_1 \\ \qquad \text{choose } 1 \times 1 \text{ pivot} \\ \textbf{else} \\ \qquad \text{choose } 2 \times 2 \text{ pivot} \end{array} \tag{25}$$

With this strategy, the entries of L are trivially bounded by 1 in absolute value, while the best one can say for Bunch–Parlett is that $|l_{ij}| \leq 1/\alpha \approx 1.6$ (for the elements generated by 1×1 pivots). Also, the condition number in the 2-norm of the pivots is bounded by 4.6 for Bunch–Parlett and by 2.6 for this new strategy. Notice that the change in the value of α does not affect the validity of the results in section 3.1.1, since Lemma 4 remains true for all $\alpha \leq 1$.

We are now in the position to prove that, with this modified pivoting strategy, the condition number of the factor L of the block LDL^T factorization (4) grows at most quadratically with the dimension of the factorized matrix.

THEOREM 3. *Let A be a symmetric DSTU matrix. There is a DSTU matrix B whose unit lower triangular factor computed by Gaussian elimination with complete pivoting coincides with the triangular factor of the block LDL^T factorization of A obtained using the pivoting strategy* (25)*. Therefore, the condition number of the latter triangular factor grows at most quadratically with the dimension of A.*

Proof. Without loss of generality, we may restrict ourselves to comparing the first two steps of GECP with the corresponding steps of the diagonal pivoting method. If the first pivot in the diagonal pivoting method is 1×1, then it applies to A the same permutations as GECP, and the entries of the first column of both triangular factors trivially coincide. Moreover, since the pivot is chosen from the diagonal, the matrix is symmetrically permuted by GECP and the Schur complements also coincide for both methods.

Now, suppose that the first pivot in the diagonal pivoting method is 2×2, say

$$\begin{pmatrix} a_{pp} & a_{pq} \\ a_{qp} & 0 \end{pmatrix}.$$

The proof for the case $a_{pp} = 0$ is completely analogous. Then, if we denote by P_{ij} the permutation which interchanges rows i and j, the diagonal pivoting method permutes the matrix A into $P_{2q}P_{1p}\ A\ P_{1p}P_{2q}$ in order to place the 2×2 pivot matrix in the upper left corner. GECP, on the other hand, would permute A to $P_{1p}AP_{1q}$ to place a_{pq} in the upper left corner of the matrix. However, it will be convenient for our purpose to apply some additional permutations: applying P_{2q} on the rows places the zero entry a_{qq} in the $(2,1)$ position. Applying P_{2q} and P_{2p} on the columns places the entry $a_{qp} = a_{pq}$ in the $(2,2)$ position and reorders the entries in such a way that, if we rename

$$M_1 = P_{2q}P_{1p}\ A\ P_{1p}P_{2q}, \qquad M_2 = P_{2q}P_{1p}AP_{1q}P_{2q}P_{2p},$$

both matrices M_1 and M_2 are identical, except for the first two columns, which are switched. Now, denote by m_{ij} the (i,j) element of the symmetric matrix M_1 (recall that $m_{22} = 0$, and m_{12} is the entry of M_1 with largest absolute value). Then the diagonal pivoting method computes the entries of the first two columns of L as

$$\begin{aligned} l_{i1} &= \frac{m_{i2}}{m_{12}}, \quad i = 3, \ldots, n, \\ l_{i2} &= \frac{m_{i1}m_{12} - m_{i2}m_{11}}{m_{12}^2}, \quad i = 3, \ldots, n, \end{aligned}$$

and the entries of the $(n-2) \times (n-2)$ Schur complement are

(26)
$$(B - CE^{-1}C^T)_{ij} = m_{ij} + \frac{-m_{i2}m_{12}m_{1j} - m_{i1}m_{12}m_{2j} + m_{i2}m_{11}m_{2j}}{m_{12}^2}, \quad i,j = 3, \ldots, n.$$

We will prove that the first two steps of Gaussian elimination on M_2 produce exactly the same two rows of L and the same $(n-2) \times (n-2)$ Schur complement. The first step of Gaussian elimination on M_2 trivially produces a first column with a zero in the first position, and $l_{i1} = m_{i2}/m_{12},\ i = 3, \ldots, n$ as above. The $(n-1) \times (n-1)$ resulting Schur complement has the form

(27)
$$\left(\begin{array}{c|ccc} m_{12} & \cdots & m_{2j} & \cdots \\ \hline \vdots & \vdots & \vdots & \vdots \\ m_{i1} - \dfrac{m_{i2}m_{11}}{m_{12}} & \vdots & m_{ij} - \dfrac{m_{i2}m_{1j}}{m_{12}} & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{array}\right).$$

In the second step, GECP looks for the entry with largest absolute value in this matrix. We claim this entry is again m_{12}; if not, there must be some new entry, which was not in M_2, larger than m_{12} in absolute value. Any entry of the Schur complement (27) vanishes whenever it is the result of a subtraction with nonzero operands, so the only possibly new elements are quotients $-(m_{i2}m_{1j})/(m_{12})$. These quotients are still monomials with coefficient ± 1 in the two diagonal entries of $\mathcal{D}$ corresponding to the indices i and j before A was permuted to M_2. In any case, since both i, j are different from either 1 or 2, both $\widetilde{d}_i$ and $\widetilde{d}_j$ are different from d_p and d_q. Consequently, the absolute value of any new element in (27) is strictly smaller than the maximum $|m_{12}| = |d_p d_q|$.

Hence, no permutation is needed in the second step of GECP on M_2. This step produces a second column for L with entries

$$l_{i2} = \frac{m_{i1} - \dfrac{m_{11}m_{i2}}{m_{12}}}{m_{12}} = \frac{m_{i1}m_{12} - m_{11}m_{i2}}{m_{12}^2}$$

which coincide with the entries l_{i2} above, replacing i by j (recall that $m_{ij} = m_{ji}$). Finally, the $(n-2) \times (n-2)$ Schur complement computed by GE in this second step has entries

$$\left(m_{ij} - \frac{m_{i2}m_{1j}}{m_{12}}\right) - \frac{\left(m_{1i} - \dfrac{m_{11}m_{i2}}{m_{12}}\right) m_{2j}}{m_{12}} \ .$$

A straightforward computation shows the equality of this formula with (26). This proves that, as claimed, two steps of Gaussian elimination on M_2 produce the same two columns for L as one 2×2 step of the diagonal pivoting method on M_1. Furthermore, the remaining $(n-2) \times (n-2)$ Schur complement is also the same.

Therefore, repeating the argument for the subsequent steps of both decompositions, we obtain that the factor L of the symmetric decomposition (4) of A is equal to the lower triangular factor of the LDU decomposition of a nonsymmetric matrix $B = AQ$ for an appropriate permutation matrix Q. Notice that B is DSTU if A is DSTU, since

$$B = D_L \widetilde{Z} D_R,$$

with $D_L = D$, $D_R = Q^T D Q$, $\widetilde{Z} = ZQ$, and the latter is trivially TU. The bound on the condition number of L follows trivially from Theorem 10.2 in [11]. ∎

The proof of Theorem 3 relies somewhat indirectly on [11, Theorem 10.2]. Trying a more direct path, with a proof analogous to the one of [11, Theorem 10.2] is unfeasible, since we lack one of the essential ingredients, namely the property of Gaussian elimination that the elements of any minor of L are a quotient of minors of the original matrix A and, therefore, the elements of L^{-1} are quotients of minors of A. This no longer holds for the LDL^T factorization, as shown by the 3×3 example in section 2.

3.2. Total signed compound matrices.

DEFINITION 3. *Let $\mathcal{S}$ be a set of matrices with given sparsity and sign pattern, i.e., all matrices in $\mathcal{S}$ have their nonzero entries in the same position and with the same sign. The set $\mathcal{S}$ is* total signed compound (TSC) *if, for every $A \in \mathcal{S}$ and for every square submatrix M of A, the Laplace expansion*

$$\det M = \sum_{\pi} [\mathrm{sign}(p) m_{1,\pi_1} m_{2,\pi_2} \dots m_{s,\pi_s}] \tag{28}$$

of the determinant of M is either a sum of monomials of like sign, with at least one nonzero monomial, or identically zero (i.e., no nonzero monomial appears in the expression).

There are well-known classes of pattern matrices among the TSC, provided their particular sign distribution conforms to the TSC condition. Two such examples are,

for instance, the tridiagonal pattern

$$\begin{pmatrix} + & + & & & \\ & + & - & + & \\ & + & + & + & \\ & & + & - & + \\ & & & + & + \end{pmatrix}$$

and the arrowhead pattern

$$\begin{pmatrix} + & + & + & + & + \\ + & - & & & \\ + & & - & & \\ + & & & - & \\ + & & & & - \end{pmatrix}.$$

TSC matrices are rather sparse (there are at most $3n-2$ nonzero entries in an $n \times n$ TSC matrix), and there are $O(n)$ algorithms for computing the determinant of an $n \times n$ TSC matrices. Moreover, such algorithms compute the determinant to high relative accuracy, since, due to the TSC property, no cancellation occurs in the calculation (recall that the determinant is determined to high relative accuracy, since it is subtraction-free). The $O(n)$ cost is achieved by making use of an alternative, constructive definition of TSC matrices: every TSC matrix can be constructed starting from a 1×1 nonzero matrix and repeatedly applying four construction rules (see [1, 11] for more details). If we restrict ourselves to *symmetric* TSC matrices, one can prove that only three construction rules are needed.

THEOREM 4. *Every TSC symmetric matrix can be obtained by starting with a* 1×1 *nonzero matrix and applying the following three construction rules repeatedly in some order:*

1. *If A is symmetric and TSC, then permuting two rows and the corresponding columns, or multiplying by -1 one row and the corresponding column, leaves A symmetric TSC.*
2. *If A_1 and A_2 are symmetric TSC matrices, then so is the direct sum*
$$\begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}.$$
3. *If the $n \times n$ $\widetilde{A}$ is symmetric and TSC, with $\widetilde{a}_{ii} \neq 0$, then so is the $(n+1) \times (n+1)$ matrix A obtained as follows:*
$$A = \left(\begin{array}{ccccccc|c} & & & & & & & 0 \\ & & & & & & & \vdots \\ & & & & & & & 0 \\ & & & \widetilde{A} & & & & a_{i,n+1} \\ & & & & & & & 0 \\ & & & & & & & \vdots \\ & & & & & & & 0 \\ \hline 0 & \dots & 0 & a_{i,n+1} & 0 & \dots & 0 & a_{n+1,n+1} \end{array}\right),$$
where we can also set $\widetilde{a}_{ii}$ to zero. The new possibly nonzero entries $a_{i,n+1}$ and $a_{n+1,n+1}$ must be chosen so that the two monomials in the minor $a_{n+1,n+1}\widetilde{a}_{i,i} - a_{i,n+1}a_{n+1,i}$ have the same sign (or are zero).

These rules will allow us to inexpensively generate TSC matrices in section 5.

3.2.1. Accurate block LDL^T factorization of symmetric TSC matrices. Our interest in TSC matrices stems from the fact that all their minors can be computed without subtraction and therefore with no cancellation. According to Theorem 1, any intermediate quantity in the process of the block LDL^T factorization is a quotient of minors (or just a minor) of the original matrix. Although the standard formulas for the diagonal pivoting method may require subtraction and therefore lead to cancellation, that subtraction can be avoided if the corresponding element of L or of the Schur complement is recomputed as a quotient of minors of the original TSC matrix. This can be done using the formulas in Lemma 1 for the Schur complements and the formulas in Lemma 2 for the elements of L. Moreover, the lack of cancellation implies that these minors are computed to high relative accuracy.

The argument above amounts to proving the following theorem.

THEOREM 5. *Algorithm* 2 *computes all entries of the L and D factors of the block LDL^T factorization of a symmetric TSC matrix to high relative accuracy, i.e.,*

$$\frac{|\widehat{l}_{ij} - l_{ij}|}{|l_{ij}|} = O(\epsilon), \quad \frac{|\widehat{d}_{ij} - d_{ij}|}{|d_{ij}|} = O(\epsilon),$$

where $\widehat{L}$ and $\widehat{D}$ are the factors computed in floating point arithmetic by Algorithm 2 *and L and D are the the exact factors which the diagonal pivoting method would compute in exact arithmetic choosing the pivots with the same dimensions and positions as those chosen in floating point arithmetic to compute $\widehat{L}$ and $\widehat{D}$.*

We now write the pseudocode for the corresponding algorithm. Of course, recomputing represents an overhead cost, since every $s \times s$ minor costs $O(s)$ operations instead of the $O(1)$ operations for the standard formula (any submatrix of a TSC matrix is trivially TSC). Therefore, the following modification of the diagonal pivoting method can cost in the worst case as much as $O(n^4)$ arithmetic operations, the same asymptotic order of the algorithm computing nonsymmetric RRDs of TSC matrices in [11] (see [11, Theorem 7.2, p. 60]). For more on this question, see the experiment below at the end of section 5.2.

ALGORITHM 2. BLOCK LDL^T FACTORIZATION OF A SYMMETRIC TSC MATRIX A.

Input: symmetric $n \times n$ TSC matrix A

Output: unit lower triangular matrix L, block diagonal matrix D with 1×1 and 2×2 diagonal blocks, permutation matrix P such that $PAP^T = LDL^T$.

1. **for** $i = 1$ **to** n
2. choose pivot according to Bunch–Parlett pivoting strategy
3. **if** 1×1 pivot, a_{ii}
4. $D_{ii} = a_{ii}$
5. **for** $j = i + 1$ **to** n
6. $l_{ji} = a_{ji}/a_{ii}$
7. **endfor**
8. **for** $j = i + 1$ **to** n
9. **for** $k = i + 1$ **to** n
10. $a_{jk} = a_{jk} - \dfrac{a_{ji}a_{ik}}{D_{ii}}$
11. (*) If the last subtraction has two nonzero operands with the same sign, recompute a_{jk} as the quotient of two minors of A according to formula (13) in Lemma 1

endfor
endfor
elseif 2×2 pivot, $\begin{pmatrix} a_{ii} & a_{i,i+1} \\ a_{i+1,i} & a_{i+1,i+1} \end{pmatrix}$
$D_{ii} = a_{ii}, D_{i,i+1} = D_{i+1,i} = a_{i,i+1}, D_{i+1,i+1} = a_{i+1,i+1}$
for $j = i + 1$ **to** n
$dpiv = a_{ii}a_{i+1,i+1} - a_{i,i+1}^2$
(*) If this subtraction has two nonzero operands with the same sign, recompute $dpiv$ as the quotient of two minors of A according to formula (14) in Lemma 1
$l_{ji} = \frac{a_{ji}a_{i+1,i+1}}{dpiv} - \frac{a_{j,i+1}a_{i,i+1}}{dpiv}$
(*) If the last subtraction has two nonzero operands with the same sign, recompute l_{ji} as the quotient of two minors of A according to formula (15) in Lemma 2
endfor
for $j = i + 2$ **to** n
$l_{j,i+1} = -\frac{a_{j,i+1}a_{i,i+1}}{dpiv} + \frac{a_{j,i+1}a_{ii}}{dpiv}$
(*) If the last subtraction has two nonzero operands with the same sign, recompute $l_{j,i+1}$ as the quotient of two minors of A according to formula (15) in Lemma 2
endfor
for $j = i + 1$ **to** n
for $k = i + 1$ **to** n
$a_{jk} = a_{jk} - \frac{a_{jp}a_{qq}a_{pk}}{dpiv} - \frac{a_{jq}a_{pq}a_{pk}}{dpiv} - \frac{a_{jp}a_{pq}a_{qk}}{dpiv} + \frac{a_{jq}a_{pp}a_{qk}}{dpiv}$
(*) If the last subtraction has two nonzero operands with the same sign, recompute a_{jk} as the quotient of two minors of A according to formula (13) in Lemma 1
endfor
endfor
endif
endfor

4. From LDL^T to RRD. Once we have a block LDL^T factorization, we have seen in (5) how to obtain a symmetric RRD by Givens diagonalization. It is not hard to show that, since L is computed with small elementwise error and $X = P^T LQ$ is, up to permutations, the result of a floating point Givens transformation (see, e.g., [6, Lemma 3.1]), the computed X satisfy (7). However, we will prove a tighter bound in Theorem 6 below, namely that X is computed with *columnwise* small relative errors. Note also that X and L have the same condition number, so if L is well-conditioned, then so is X (this is guaranteed, for instance, for DSTU matrices, according to Theorem 3).

4.1. Error analysis. We present here the error analysis showing that the block LDL^T factorization, followed by Givens diagonalization, leads to RRDs satisfying the requirement (7) for accurately computing eigenvalues and eigenvectors via the signed SVD method. We make no distinction between DSTU and TSC matrices, since the error analysis is valid for any matrix such that its block LDL^T decomposition can be

computed with small componentwise error as in (6). The analysis is related to the one in [15] and uses some results appearing in [15]. To be more precise we need some notation: we assume the conventional model for floating point arithmetic,

$$\texttt{fl}(a \odot b) = (a \odot b)(1+\delta), \tag{29}$$

where a and b are real floating point numbers, $\odot \in \{+,-,\times,/\}$, and $|\delta| \leq \epsilon$, where ϵ is the machine precision. Moreover, we assume that neither overflow nor underflow occur. Also, for each $k > 0$ we set

$$\gamma_k = \frac{k\epsilon}{1-k\epsilon}, \tag{30}$$

and as in [18, section 3.4], we denote by θ_k any positive quantity bounded by γ_k. Finally, given a real symmetric 2×2 matrix, we write the Jacobi orthogonal diagonalization procedure as

$$\begin{pmatrix} a & c \\ c & b \end{pmatrix} = \begin{pmatrix} cs & sn \\ -sn & cs \end{pmatrix} \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} \begin{pmatrix} cs & -sn \\ sn & cs \end{pmatrix}, \tag{31}$$

with $\lambda_1 = a - ct,\ \lambda_2 = b + ct$, where

$$t = \frac{\operatorname{sign}(\zeta)}{|\zeta| + \sqrt{1+\zeta^2}} \qquad \text{for} \qquad \zeta = \frac{b-a}{2c} \tag{32}$$

and

$$cs = \frac{1}{\sqrt{1+t^2}}, \qquad sn = cs \cdot t. \tag{33}$$

The main result in this section, written with this notation, is the following.

THEOREM 6. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix and let $\widehat{L}, \widehat{D}$ be the computed factors of a block LDL^T factorization of A obtained through the diagonal pivoting method using the Bunch–Parlett pivoting strategy* (11) *(i.e., $\alpha = (1+\sqrt{17})/8$). Suppose that $\widehat{L}, \widehat{D}$ have been computed with small componentwise relative error*

$$\begin{aligned} \widehat{l}_{ij} &= l_{ij}(1+\theta_{K_L}^{(ij)}), \qquad i,j = 1,\dots,n, \\ \widehat{d}_{ij} &= l_{ij}(1+\theta_{K_D}^{(ij)}), \qquad i,j = 1,\dots,n, \end{aligned} \tag{34}$$

for appropriate constants $K_L, K_D > 0$, and let $\widehat{X}, \widehat{\Delta}$ (resp., X, Δ) be the computed (resp., exact) factors of a symmetric RRD obtained by Givens diagonalization (5) *in floating point (resp., in exact arithmetic) using formulas* (31)–(33). *Then*

$$\frac{|\widehat{\Delta}_{jj} - \Delta_{jj}|}{|\Delta_{jj}|} \leq 4\,\frac{1+\alpha}{1-\alpha}\,\gamma_{K_D+29}, \qquad j = 1,\dots,n, \tag{35}$$

and

$$\frac{\|\widehat{X}(:,j) - X(:,j)\|_2}{\|X(:,j)\|_2} \leq \sqrt{2nC^2+1}\ \gamma_M, \qquad j = 1,\dots,n, \tag{36}$$

where C is

$$C = \frac{1}{1-\alpha} + O(\epsilon).$$

and

$$M = \max\{48K_{_L} + 141, K_{_L} + 48K_{_D} + 143\}. \tag{37}$$

To prove it we will use the fact that the computed diagonalizing transformation is close entrywise to the exact one. The following result is taken from [15, Appendix A.3].

LEMMA 5 (see [15]). *Let*

$$\widetilde{A} = \begin{pmatrix} \widetilde{a} & \widetilde{c} \\ \widetilde{c} & \widetilde{b} \end{pmatrix} = \begin{pmatrix} a(1+\delta_a) & c(1+\delta_c) \\ c(1+\delta_c) & b(1+\delta_b) \end{pmatrix}$$

be a matrix of real floating point numbers, with $\max\{|\delta_a|, |\delta_b|, |\delta_c|\} \le \gamma_k$ *and* $\alpha|\widetilde{c}| \ge \max\{|\widetilde{a}|, |\widetilde{b}|\}$. *Let*

$$A = \begin{pmatrix} a & c \\ c & b \end{pmatrix},$$

with eigenvalues $\lambda_1 \ge \lambda_2$, *and orthonormal eigenvectors* $v_1 = [cs, -sn]^T$ *and* $v_2 = [sn, cs]$. *Let* $\widehat{\lambda}_1, \widehat{\lambda}_2, \widehat{cs}$ *and* $\widehat{sn}$ *be the versions of* λ_1, λ_2, cs *and* sn *computed in floating point arithmetic for* $\widetilde{A}$ *according to formulas* (31)–(33). *If*

$$4\sqrt{2}\,\frac{1+\alpha}{1-\alpha}\gamma_{k+29} \le 1 \qquad \textit{and} \qquad \gamma_{141+48k} \le 1,$$

then

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} \le 4\,\frac{1+\alpha}{1-\alpha}\gamma_{k+29}, \quad i = 1, 2, \tag{38}$$

and

$$\widehat{cs} = cs(1 + \theta_{16k+113}), \qquad \widehat{sn} = cs(1 + \theta_{48k+141}). \tag{39}$$

Proof of Theorem 6. First, we may assume that $P = I$, since no error is introduced by the pivoting permutations. Let $\widehat{Q}$ be the computed orthogonal matrix diagonalizing $\widehat{D}$; i.e., if $\widehat{D} = \operatorname{diag}(\widehat{D}_1, \ldots, \widehat{D}_r)$ with $D_k \in \mathbb{R}^{s_k \times s_k}$, $s_k = 1$ or 2, $k = 1, \ldots, r$, then $\widehat{Q} = \operatorname{diag}(\widehat{Q}_1, \ldots, \widehat{Q}_r)$ with $Q_k \in \mathbb{R}^{s_k \times s_k}$, $k = 1, \ldots, r$. The 1×1 blocks of $\widehat{Q}_k$ are equal to 1, and each 2×2 block

$$\widehat{Q}_k = \begin{pmatrix} \widehat{cs} & -\widehat{sn} \\ \widehat{sn} & \widehat{cs} \end{pmatrix} \tag{40}$$

is the version computed in floating point arithmetic of the Jacobi rotation which would diagonalize the 2×2 block $\widehat{D}_k$ in exact arithmetic. Analogously, $Q = \operatorname{diag}(Q_1, \ldots, Q_r)$, where

$$Q_k = \begin{pmatrix} cs & -sn \\ sn & cs \end{pmatrix}$$

is the exact Jacobi rotation diagonalizing the diagonal block D_k of D. For those columns j corresponding to a pivot with $s_k = 1$, we have $\widehat{\Delta}_{jj} = \widehat{d}_{jj}$, $\Delta_{jj} = d_{jj}$ and

$\widehat{X}(:,j) = \widehat{L}(:,j)$, $X(:,j) = L(:,j)$, so (35) and (36) are trivially satisfied. Therefore, only the columns corresponding to 2×2 pivots must be considered. Let the jth and $(j+1)$st be two such columns. First, inequality (35) follows directly from applying Lemma 5 in our setting, i.e., taking $\widetilde{A}$, A, λ_1, λ_2 , and k to be equal, respectively, to $\widehat{D}$, D, Δ_{jj}, $\Delta_{j+1,j+1}$, and $K_{_D}$. With this choice, inequality (38) reduces to (35). To prove (36), note that

$$X = LQ, \qquad \widehat{X} = \mathtt{fl}(\widehat{L}\widehat{Q}),$$

where $\mathtt{fl}(expr)$ denotes the computed result in finite precision of expression $expr$. Reading these identities entrywise for columns j and $j+1$, we get

$$\widehat{X}(i,j) = \begin{cases} 0 & \text{if} \quad i < j\ , \\ \widehat{cs} & \text{if} \quad i = j\ , \\ \widehat{sn} & \text{if} \quad i = j+1\ , \\ \mathtt{fl}(\widehat{l}_{ij}\,\widehat{cs} + \widehat{l}_{i,j+1}\,\widehat{sn}) & \text{if} \quad i > j+1\ , \end{cases} \tag{41}$$

$$\widehat{X}(i,j+1) = \begin{cases} 0 & \text{if} \quad i < j\ , \\ -\widehat{sn} & \text{if} \quad i = j\ , \\ \widehat{cs} & \text{if} \quad i = j+1\ , \\ \mathtt{fl}(-\widehat{l}_{ij}\widehat{sn} + \widehat{l}_{i,j+1}\widehat{cs}) & \text{if} \quad i > j+1 \end{cases}$$

and the same equalities without hats for the entries of X. Therefore,

$$||\widehat{X}(:,j)-X(:,j)||_2^2 = (\widehat{cs}-cs)^2+(\widehat{sn}-sn)^2+\sum_{i=j+2}^{n}[\mathtt{fl}(\widehat{l}_{ij}\widehat{cs}+\widehat{l}_{i,j+1}\widehat{sn})-(l_{ij}cs+l_{i,j+1}sn)]^2.$$

Using (34), (29), and (39), we may write

$$\begin{aligned}\mathtt{fl}(\widehat{l}_{ij}\widehat{cs} + \widehat{l}_{i,j+1}\widehat{sn}) &= \Big[l_{ij}\,cs\,(1+\theta_{K_{_L}}^{(i,j)})\,(1+\theta_{16K_{_D}+113})\,(1+\delta_1) \\ &\qquad + l_{i,j+1}\,sn\,(1+\theta_{K_{_L}}^{(i,j+1)})\,(1+\theta_{48K_{_D}+141})\,(1+\delta_2)\Big]\,(1+\delta_3) \\ &= l_{ij}\,cs\,(1+\theta_{K_{_L}+16K_{_D}+115}) + l_{i,j+1}\,sn\,(1+\theta_{K_{_L}+48K_{_D}+143}).\end{aligned}$$

Hence, again using (39),

$$\begin{aligned}\|\widehat{X}(:,j) - X(:,j)\|_2^2 &= (cs\,\theta_{16K_{_D}+113})^2 + (sn\,\theta_{48K_{_D}+141})^2 + \\ &\qquad + \sum_{i=j+2}^{n}\Big(l_{ij}\,cs\,\theta_{K_{_L}+16K_{_D}+115} + l_{i,j+1}\,sn\,\theta_{K_{_L}+48K_{_D}+143}\Big)^2 \\ &\le (\gamma_{48K_{_L}+141})^2 + 2n\,(\gamma_{K_{_L}+48K_{_D}+143})^2\max\{|l_{ij}|^2, |l_{i,j+1}|^2\},\end{aligned}$$

where we have used the monotonicity of γ_k in k. At this point, we must observe that, although the the Bunch–Parlett strategy ensures that the entries of the *computed* $\widehat{L}$ satisfy $|\widehat{l}_{ik}| \le 1/(1-\alpha)$ for all i,k, this may not be true for the entries l_{ik} of the *exact* L. The entrywise bound (34), however, implies (after some calculations) that

$|l_{ik}| \leq C$ for a constant C which is equal to $1/(1-\alpha)$ up to first order terms[4] in ϵ. Therefore,

$$\|\widehat{X}(:,j) - X(:,j)\|_2 \leq \sqrt{1+2nC^2}\,\gamma_M$$

with M given by (37), which leads trivially to (36), since $\|X(:,j)\|_2^2 \geq cs^2 + sn^2 \geq 1$. □

5. Numerical experiments. We have performed extensive numerical tests which confirm the correctness of our algorithms. All of them were done in MATLAB 5.3 using an AMD Athlon (tm) XP 2000+ processor with IEEE arithmetic.

We have used as a reference the eigenvalues and eigenvectors computed using Maple's variable-precision arithmetic available from the Symbolic Math Toolbox of MATLAB through the command `vpa`. For each matrix A, the "exact" eigendecomposition is obtained using MATLAB's usual command `eig` (i.e., with the QR algorithm) but setting the variable `digits`, which specifies the number of significant decimal digits used by Maple to $18+d$ if the condition number of A is $O(10^d)$. We denote by λ_i and q_i the eigenvalues and eigenvectors computed in this way and by $\widehat{\lambda}_i$ and $\widehat{q}_i$ those computed via the signed SVD method implemented in MATLAB. Therefore, $\widehat{\lambda}_i, \widehat{q}_i$ are computed in double precision arithmetic, i.e., $\epsilon \approx 2.2 \cdot 10^{-16}$. The initial RRD factorization is implemented in MATLAB, using Algorithm 1 for DSTU matrices and Algorithm 2 for TSC matrices.

We analyzed the following quantities:

1. the maximum relative error in eigenvalues:

$$e_\lambda = \max_i \left| \frac{\lambda_i - \widehat{\lambda}_i}{\lambda_i} \right|; \tag{42}$$

2. a control quantity for eigenvalues:

$$\vartheta_\lambda = \frac{e_\lambda}{\kappa\epsilon}, \tag{43}$$

   where ϵ is the machine precision, and $\kappa = \kappa(R')\,\kappa(X)$ as in (8)—this quantity is expected to be roughly $O(1)$ in the experiments;
3. the maximum relative error in the eigenvectors:

$$e_q = \max_i ||\widehat{q}_i - q_i||_2; \tag{44}$$

4. a control quantity for eigenvectors:

$$\xi_q = \max_i \frac{||\widehat{q}_i - q_i||_2\,\mathrm{relgap}^*(\widehat{\lambda}_i)}{\kappa\epsilon} \tag{45}$$

   with κ as above and relgap* defined as in (9). Again, ξ_q should be $O(1)$ for the experiments to confirm our analysis.

[4] See [15] for a more detailed analysis, which proves that

$$C = \frac{1}{(1-\alpha)(1-\gamma_{g(\alpha)})}, \qquad g(\alpha) = \left(32\left(\frac{1+\alpha}{1-\alpha}\right)^2 + 196\,\frac{1+\alpha}{1-\alpha}\right) K_D.$$

5.1. Diagonally scaled totally unimodular matrices. We generated TU nonsingular matrices of sizes $6, 8, 10$, and 12. We were unable to generate matrices of larger dimension due to the high cost of the generating routine: it generates TU matrices recursively, starting from a TU matrix of size 1, i.e., either $-1, 1$, or 0. Given a generated matrix of size s, the algorithm constructs a $s+1 \times s+1$ TU matrix by adjoining a new row and column, with entries randomly chosen among $-1, 1, 0$, and checking whether all new minors containing entries from that row and column are equal to $-1, 1$, or 0. The computational cost of checking the minors is what makes the algorithm so costly.

Once we have a TU matrix, we scale it on both sides with diagonal matrices with powers of 10 on the diagonal, their condition numbers ranging from 10^5 to 10^{20}. Therefore, the corresponding DSTU matrices will have condition numbers ranging roughly from 10^{10} to 10^{40}. For each size we divide the experiments in three groups according to their condition number: condition numbers ranging from 10^{10} to 10^{20}, from 10^{20} to 10^{30}, and from 10^{30} to 10^{40}. We generate 100 matrices for each range, so the following tables reflect the results on 1200 matrices, 300 for each dimension. Table 1 shows the control quantities ϑ_λ for eigenvalues, while Table 2 shows the control quantities ξ_q for eigenvectors. For each dimension there are two columns, the left one displaying the average over the 100 tests made in that range of condition numbers and the right one displaying the largest value for the control quantity among the 100 experiments.

TABLE 1
Statistical data for accuracy in eigenvalues of DSTU matrices: ϑ_λ.

	$n=6$		$n=8$		$n=10$		$n=12$	
$\kappa(A) = O(10^d)$	Mean	Max	Mean	Max	Mean	Max	Mean	Max
$10 \le d \le 20$	1.412	6.689	1.746	32.34	1.879	19.14	1.425	9.310
$20 \le d \le 30$	1.460	16.34	1.652	38.14	1.432	13.49	1.696	45.45
$30 \le d \le 40$	1.699	26.65	1.338	11.34	1.157	3.949	1.719	33.02

TABLE 2
Statistical data for accuracy in eigenvectors of DSTU matrices: ξ_q.

	$n=6$		$n=8$		$n=10$		$n=12$	
$\kappa(A) = O(10^d)$	Mean	Max	Mean	Max	Mean	Max	Mean	Max
$10 \le d \le 20$	0.508	2.653	0.508	1.886	0.579	1.989	0.605	2.364
$20 \le d \le 30$	0.502	1.914	0.518	1.716	0.623	2.214	0.603	1.928
$30 \le d \le 40$	0.447	1.884	0.582	2.795	0.571	2.840	0.621	2.697

5.2. Total signed compound matrices. We generated TSC matrices of sizes $10, 20, 40$, and 60 by starting from a nonzero 1×1 matrix and repeatedly applying rules 2 and 3 in Theorem 4. Rule 2 was applied with a probability of 5%, choosing as A_2 one of the blocks of the TSC matrix A_1 computed in the previous stage. Otherwise, rule 3 was applied, generating the new quantities $a_{i,n+1}$, $a_{n+1,n+1}$ with MATLAB's `rand` command. Whenever rule 3 was employed, the diagonal entry $\widetilde{a}_{ii}$ was set to zero with a probability of 20%. Finally, large condition numbers were induced by scaling the resulting matrices on both sides with ill-conditioned diagonal matrices, exactly as in the experiment for DSTU matrices. Notice that, since the scaling matrices were positive, the sign pattern of the matrix does not change under scaling. Again, 1200 matrices were generated, 100 for each dimension and each range of condition numbers.

The results are summarized in Tables 3 and 4.

TABLE 3
Statistical data for accuracy in eigenvalues of TSC matrices: ϑ_λ.

	$n = 10$		$n = 20$		$n = 40$		$n = 60$	
$\kappa(A) = O(10^d)$	Mean	Max	Mean	Max	Mean	Max	Mean	Max
$10 \le d \le 20$	1.446	10.024	1.449	4.196	1.940	8.802	2.280	9.639
$20 \le d \le 30$	1.332	6.579	2.170	38.68	2.033	5.172	2.278	9.528
$30 \le d \le 40$	1.362	5.973	1.591	7.411	2.841	44.70	2.502	9.583

TABLE 4
Statistical data for accuracy in eigenvectors of TSC matrices: ξ_q.

	$n = 10$		$n = 20$		$n = 40$		$n = 60$	
$\kappa(A) = O(10^d)$	Mean	Max	Mean	Max	Mean	Max	Mean	Max
$10 \le d \le 20$	0.682	3.044	0.843	2.987	1.292	3.342	1.418	3.641
$20 \le d \le 30$	0.717	7.438	0.889	4.215	1.294	3.034	1.405	3.665
$30 \le d \le 40$	0.800	3.768	0.893	3.386	1.265	2.802	1.471	3.672

As can be seen from the tables, the results confirm our theoretical predictions. In parallel, we also computed eigenvalues and eigenvectors of the test matrices with MATLAB's `eig` command. As expected, the relative errors were huge, providing no correct digit in the smaller eigenvalues.

We conclude with an experiment to estimate the actual computational cost of the LDL^T factorization for symmetric TSC matrices: we randomly generate one hundred symmetric TSC matrices for each size from 10 to 100 in steps of ten, i.e., we generate one hundred 10×10 matrices, one hundred 20×20 matrices, one hundred 30×30 matrices and so on, i.e., one thousand test matrices in all. For each matrix we compute an LDL^T factorization using Algorithm 2, and we record the number of flops employed by the factorization procedure. Each star in Figure 1 corresponds to a given size and represents the arithmetic mean of the one hundred data obtained for that size, plotted in a log-log scale, with the logarithm of the size n of the matrix on the horizontal axis. The solid line corresponds to `flops`$= n^4$, and the dashed line to `flops`$= n^3$. As can be seen in the figure, the cost seems to be somewhere in between. However, since we have no estimation of the constants involved in the big-oh, it is hard to draw any specific conclusion, other than that the cost seems not to be too high.

Appendix A. Proofs of results in section 2.

Underlying the results in section 2 is one of the most useful properties of Schur complements, usually known as the *quotient property* (see, e.g., [5, section 2]).

LEMMA 6. *Let M be any square matrix, partitioned as*

$$M = \begin{pmatrix} B & * \\ * & * \end{pmatrix}, \qquad \textit{with} \qquad B = \begin{pmatrix} B_1 & * \\ * & * \end{pmatrix},$$

where B and B_1 are square nonsingular. Let $\mathcal{C}_1^B$ (resp., $\mathcal{C}_1^M$) be the Schur complement of B_1 in B (resp., in M). Then the Schur complement of B in M is equal to the Schur complement of $\mathcal{C}_1^B$ in $\mathcal{C}_1^M$.

With this result one can easily prove Lemma 1.

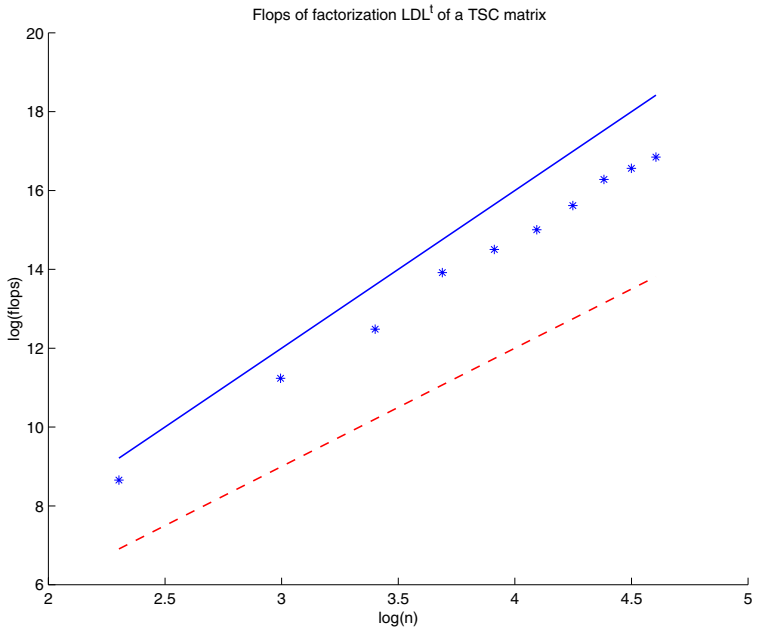

FIG. 1. *Computational cost of LDL^T factorization for symmetric TSC matrices of sizes ranging from* 10 *to* 100.

Proof of Lemma 1. We will prove (a) by induction on k. If $k = 1$, the elements of $\mathcal{C}_1$ are

$$\mathcal{C}_1(i,j) = a_{i+1,j+1} - \frac{a_{i+1,1}\, a_{1,j+1}}{a_{11}} = \frac{a_{i+1,j+1}\, a_{11} - a_{i+1,1}\, a_{1,j+1}}{a_{11}}$$

$$= \frac{\det A([1, i+1], [1, j+1])}{\det A([1], [1])},$$

which is just (13) with $k = 1$. Now, suppose that (13) is true for some $k \in \{1, 2 \ldots n\}$. We will show that then it is also true for $k + 1$. According to Lemma 6, the Schur complement $\mathcal{C}_{k+1}$ of A_{k+1} in A is the result of taking two successive Schur complements: first the Schur complement $\mathcal{C}_k$ of A_k in A and then the Schur complement of the (1, 1) entry in $\mathcal{C}_k$. We know from the induction hypothesis that

$$\mathcal{C}_k(i,j) = \frac{\det A([1:k, k+i], [1:k, k+j])}{\det A([1:k], [1:k])}.$$

Substituting this into the formula

$$\mathcal{C}_{k+1}(i,j) = \mathcal{C}_k(i+1, j+1) - \frac{\mathcal{C}_k(i+1,1)\mathcal{C}_k(1,j+1)}{\mathcal{C}_k(1,1)}$$

for the elements of $\mathcal{C}_{k+1}$ leads to

$$\mathcal{C}_{k+1}(i,j) = \frac{\begin{vmatrix} \det A([1:k, k+i+1], [1:k, k+j+1]) & \det A([1:k, k+1], [1:k, k+j+1]) \\ \det A([1:k, k+i+1], [1:k, k+1]) & \det A([1:k, k+1], [1:k, k+1]) \end{vmatrix}}{\det A([1:k], [1:k])\, \det A([1:k, k+1], [1:k, k+1])}.$$

It suffices to apply Sylvester's identity [19, p. 22] to the numerator to obtain (13) with $k+1$ instead of k.

Once (a) has been proved, all elements of $\mathcal{C}_k([i_1,\dots,i_s],[j_1,\dots,j_s])$ can be written as quotients with the same denominator $d_k = \det A([1:k],[1:k]$. Hence, the submatrix can be written as $(1/d_k)\,M$, where, for each $l,m \in \{1,\dots,s\}$, the element (l,m) of M is $\det A([1:k,k+i_l],[1:k,k+j_m])$. Applying again Sylvester's formula to M proves part (b). ∎

Proof of Theorem 1. The proof is similar to the one of Lemma 5.1 in [11, p. 52]. The entries of D are either entries of A or entries of a Schur complement of A. Hence, by Lemma 1, any entry of D is either an entry of A or a quotient of minors of A. Now consider an entry l_{ij} of L generated by a 1×1 pivot. Then l_{ij} is a quotient of two elements of the corresponding Schur complement of A, and since both elements have been created at the same stage of the factorization algorithm, by part (a) of Lemma 1 they are quotients of the form (13) with the same denominator. Hence, both denominators cancel out in the quotient and l_{ij} is a quotient of minors of A. The argument is similar for the entries of L generated by 2×2 pivots, using part (b) of Lemma 1 instead of part (a). ∎

Proof of Lemma 2.

We distinguish the cases $s_k = 1$ and $s_k = 2$. If $s_k = 1$, then $n_k = n_{k-1}+1$ and

$$L(i,n_k) = \frac{\mathcal{C}_{k-1}(i,1)}{\mathcal{C}_{k-1}(1,1)}, \qquad i \in \{n_k+1,\dots,n\},$$

which, according to part (a) of Lemma 1, is equal, after simplifying, to

$$L(i,n_k) = \frac{\det A([1:n_{k-1},n_{k-1}+i],[1:n_{k-1},n_{k-1}+1])}{\det A([1:n_{k-1},n_{k-1}+1],[1:n_{k-1},n_{k-1}+1])} = \frac{\det A([1:n_{k-1},i],[1:n_k])}{\det A([1:n_k],[1:n_k])}.$$

If $s_k = 2$, then $n_k = n_{k-1}+2$ and, for each $i \in \{n_k+1,\dots,n\}$, we have

$$L(i,n_k-1) = \frac{\begin{vmatrix} \mathcal{C}_{k-1}(i,1) & \mathcal{C}_{k-1}(i,2) \\ \mathcal{C}_{k-1}(1,2) & \mathcal{C}_{k-1}(2,2) \end{vmatrix}}{\begin{vmatrix} \mathcal{C}_{k-1}(1,1) & \mathcal{C}_{k-1}(1,2) \\ \mathcal{C}_{k-1}(2,1) & \mathcal{C}_{k-1}(2,2) \end{vmatrix}}$$

and

$$L(i,n_k) = \frac{\begin{vmatrix} \mathcal{C}_{k-1}(1,1) & \mathcal{C}_{k-1}(2,1) \\ \mathcal{C}_{k-1}(i,1) & \mathcal{C}_{k-1}(i,2) \end{vmatrix}}{\begin{vmatrix} \mathcal{C}_{k-1}(1,1) & \mathcal{C}_{k-1}(1,2) \\ \mathcal{C}_{k-1}(2,1) & \mathcal{C}_{k-1}(2,2) \end{vmatrix}}.$$

In both cases, Sylvester's identity, combined with Lemma 1, lead to the formula in the statement. Finally, the formulas for the elements of D are trivially obtained if we use the fact that the elements of D are either elements of the original matrix or elements of some Schur complement. ∎

Acknowledgments. The authors thank Prof. Froilán M. Dopico for many helpful discussions and for suggesting the pivoting strategy in section 3.1.2. They also thank both authors of reference [15] for making it available to them, since it very much helped to simplify the presentation of section 4.1.

REFERENCES

[1] R. Brualdi and H. Ryser, *Combinatorial Matrix Theory*, Cambridge University Press, Cambridge, UK, 1991.

[2] J. R. Bunch, *Analysis of the diagonal pivoting method*, SIAM J. Numer. Anal., 8 (1971), pp. 656–680.

[3] J. R. Bunch and B. Parlett, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.

[4] J. Barlow and J. Demmel, *Computing accurate eigensystems of scaled diagonally dominant matrices*, SIAM J. Numer. Anal., 27 (1990), pp. 762–791.

[5] R. W. Cottle, *Manifestations of the Schur complement*, Linear Algebra Appl., 8 (1974), pp. 189–211.

[6] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, 2002.

[7] J. W. Demmel, *Accurate singular value decompositions of structured matrices*, SIAM J. Matrix Anal. Appl., 21 (1999), pp. 562–580.

[8] J. W. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Stat. Comp., 11 (1990), pp. 873–912.

[9] J. W. Demmel and P. Koev, *Accurate SVDs of weakly diagonally dominant M-matrices*, Numer. Math., 98 (2004), pp. 99–104.

[10] J. W. Demmel and K. Veselić, *Jacobi's method is more accurate than QR*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1204–1245.

[11] J. W. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić, and Z. Drmač, *Computing the singular value decomposition with high relative accuracy*, Linear Algebra Appl., 299 (1999), pp. 21–80.

[12] I. S. Dhillon, *A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem*, Doctoral thesis, University of California at Berkeley, Berkeley, CA, 1997.

[13] I. S. Dhillon and B. N. Parlett, *Orthogonal eigenvectors and relative gaps*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 858–899.

[14] F. M. Dopico, J. M. Molera, and J. Moro, *An orthogonal high relative accuracy algorithm for the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 301–351.

[15] F. M. Dopico and P. Koev, *Accurate symmetric rank revealing and eigendecompositions of symmetric structured matrices*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 1126–1156.

[16] Z. Drmač, *Accurate computation of the product-induced singular value decomposition with applications*, SIAM J. Numer. Anal., 35 (1998), pp. 1969–1994.

[17] K. V. Fernando and B. Parlett, *Accurate singular values and differential qd algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[18] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.

[19] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[20] A. Householder, *The Theory of Matrices in Numerical Analysis*, reprint, Dover, New York, 1975.

[21] W. Kahan, *Accurate Eigenvalues of a Symmetric Tridiagonal Matrix*, Technical Report CS-41, Department of Computer Science, Stanford University, July 1966 (revised edition in June 1968).

[22] P. Koev, *Accurate eigenvalues and SVDs of totally nonnegative matrices*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 1–23.

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

[24] J. M. Peña, *LDU decomposition with L and U well conditioned*, Electron. Trans. Numer. Anal., 18 (2004), pp. 198–208.

[25] I. Slapničar, *Accurate Symmetric Eigenreduction by a Jacobi Method*, Doctoral thesis, Fernuniversität Hagen, Hagen, Germany, 1992.

[26] I. Slapničar, *Componentwise analysis of direct factorization of real symmetric and Hermitian matrices*, Linear Algebra Appl., 272 (1998), pp. 227–275.

[27] I. Slapničar, *Highly accurate symmetric eigenvalues decomposition and hyperbolic SVD*, Linear Algebra Appl., 358 (2003), pp. 387–424.

[28] S. Vavasis, *Stable numerical algorithms for equilibrium systems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1108–1131.

[29] K. Veselić, *A Jacobi eigenreduction algorithm for definite matrix pairs*, Numer. Math., 64 (1993), pp. 241–269.

[30] K. Veselić and I. Slapničar, *Floating-point perturbations of Hermitian matrices*, Linear Algebra Appl., 195 (1993), pp. 81–116.