Tridiagonal-diagonal reduction of symmetric indefinite pairs

Tisseur, Françoise

2004

MIMS EPrint: **2006.258**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

SIAM J. MATRIX ANAL. APPL.
Vol. 26, No. 1, pp. 215–232

TRIDIAGONAL-DIAGONAL REDUCTION OF SYMMETRIC INDEFINITE PAIRS*

FRANÇOISE TISSEUR†

Abstract. We consider the reduction of a symmetric indefinite matrix pair (A, B), with B nonsingular, to tridiagonal-diagonal form by congruence transformations. This is an important reduction in solving polynomial eigenvalue problems with symmetric coefficient matrices and in frequency response computations. The pair is first reduced to symmetric-diagonal form. We describe three methods for reducing the symmetric-diagonal pair to tridiagonal-diagonal form. Two of them employ more stable versions of Brebner and Grad's pseudosymmetric Givens and pseudosymmetric Householder reductions, while the third is new and based on a combination of Householder reflectors and hyperbolic rotations. We prove an optimality condition for the transformations used in the third reduction. We present numerical experiments that compare the different approaches and show improvements over Brebner and Grad's reductions.

Key words. symmetric indefinite generalized eigenvalue problem, tridiagonalization, hyperbolic rotation, unified rotation, hyperbolic Householder reflector

AMS subject classifications. 65F15, 65F30

DOI. 10.1137/S0895479802414783

1. Introduction.

Motivation for this work comes from the symmetric polynomial eigenvalue problem (PEP)

$$(\lambda^m A_m + \lambda^{m-1} A_{m-1} + \cdots + A_0)u = 0, \tag{1.1}$$

where the A_i, $i = 0\colon m$, are $n \times n$ symmetric matrices. λ is called an eigenvalue and $u \neq 0$ is the corresponding right eigenvector. The standard way of dealing with the PEP in practice is to reformulate it as a generalized eigenvalue problem (GEP)

$$Ax = \lambda Bx, \tag{1.2}$$

of size mn. This process is called linearization, as the GEP is linear in λ. Symmetry in the problem is maintained with an appropriate choice of linearization. For example, we can take

$$A = \begin{bmatrix} 0 & \cdots & \cdots & 0 & A_0 \\ \vdots & & & A_0 & A_1 \\ \vdots & & & \vdots & \vdots \\ 0 & A_0 & & & A_{m-2} \\ A_0 & A_1 & \cdots & A_{m-2} & A_{m-1} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & \cdots & 0 & A_0 & 0 \\ \vdots & & A_0 & A_1 & \vdots \\ & & & & \\ 0 & A_0 & & \vdots & \vdots \\ A_0 & A_1 & \cdots & A_{m-2} & 0 \\ 0 & \cdots & \cdots & 0 & -A_m \end{bmatrix}$$

and $x = [u^T, \lambda u^T, \ldots, \lambda^{m-1} u^T]^T$. The resulting A and B are symmetric but not definite, and in general the pair (A, B) is indefinite.

*Received by the editors September 16, 2002; accepted for publication (in revised form) by I. S. Dhillon November 14, 2003; published electronically September 14, 2004. This work was supported by Engineering and Physical Sciences Research Council grant GR/R45079 and Nuffield Foundation grant NAL/00216/G.

http://www.siam.org/journals/simax/26-1/41478.html

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, UK (ftisseur@ma.man.ac.uk, http://www.ma.man.ac.uk/~ftisseur/).

The first step in most eigensystem computations is the reduction of the coefficient matrices, in a finite number of operations, to a simple form. Only then is an iterative procedure applied. A symmetric indefinite pair (A, B) can be reduced to Hessenberg-triangular form and the resulting generalized eigenvalue problem solved by the QZ algorithm. This approach is numerically stable, but unfortunately the reduction to Hessenberg-triangular form destroys the symmetry. Moreover, in finite precision arithmetic there is no guarantee that the set of left and right eigenvectors computed via the QZ algorithm will coincide, a property possessed by GEPs with real symmetric matrices. Also, by preserving symmetry, storage and computational costs can be reduced.

The tridiagonal-diagonal reduction of a pair (A, B) is the most compact form we can obtain in a finite number of steps. Such reductions have been proposed by Brebner and Grad [5] and by Zurmühl and Falk [26] for nonsingular B. They require nonorthogonal transformations and can be unstable. Once (A, B) is reduced to tridiagonal-diagonal form the eigenvalues and eigenvectors can be obtained by applying, for example, an HR iteration or associated iterations [5], [6], [16], [25], Uhlig's DQR algorithm [24], or, if one is interested in the eigenvalues only, Aberth's method can be used in an efficient way [1]. A robust tridiagonal-diagonal reduction is therefore of prime importance before one can consider using any of the methods cited above. We note that Garvey et al. [8] have considered a less compact form that allows the second matrix to be in tridiagonal form. One feature of their approach is that no assumption is made on the nonsingularity of the two matrices. The simultaneous tridiagonalization is convenient if one needs to solve linear systems of the form $(A - \omega B)x = b$ for many values of ω, as is required in frequency response computations [8], but it is less attractive than the tridiagonal-diagonal form for eigenvalue computations.

Three different tridiagonal-diagonal reductions for indefinite pairs (A, B) with B nonsingular are described in this paper. They all consist of two stages. The first, common to all, is the reduction of the symmetric indefinite pair (A, B) to symmetric-diagonal form (C, J) with the aid of a block LDL^T factorization of B. During the second stage, C is tridiagonalized using a sequence of congruence transformations that preserve the diagonal form of the second matrix J. Each of the three reductions proposed in this paper uses different types of transformations. These transformations are not necessarily orthogonal, so they may be unstable in finite precision arithmetic. We describe several techniques that can be used to make them more robust and to improve stability during the reduction process: in particular, pivoting and zeroing strategies in order to minimize the condition numbers of the transformations, and mixed application of hyperbolic rotations.

The paper is organized as follows. Section 2 sets up notations and definitions. It is shown that if the tridiagonal-diagonal reduction exists, it is determined up to signs by the first column of the transformation matrix. Section 3 describes the first stage of the reduction, that is, the reduction of (A, B) to symmetric-diagonal form (C, J). The description is accompanied by an error analysis. The second stage of the reduction is described in section 4. Three algorithms are proposed. The first two are an improvement over Brebner and Grad's pseudosymmetric Givens and pseudosymmetric Householder methods [5]. The third algorithm is based on transformations used to compute hyperbolic QR factorizations in indefinite least square problems [3]. Numerical comparisons of these algorithms and comparisons to Brebner and Grad's reductions are given in the last section.

2. Background material. Unless otherwise specified, $\|\cdot\|$ denotes the 2-norm. We denote by $\mathrm{diag}_q^n(\pm 1)$ the set of all $n \times n$ diagonal matrices with q diagonal elements

equal to 1 and $n-q$ equal to -1. A matrix $J \in \operatorname{diag}_q^n(\pm 1)$ for some q is called a *signature* matrix.

Let J, $\widetilde{J} \in \operatorname{diag}_q^n(\pm 1)$. A matrix $H \in \mathbb{R}^{n\times n}$ is said to be $(J,\widetilde{J})$-*orthogonal* if $H^T J H = \widetilde{J}$. Note that $(J,\widetilde{J})$-orthogonal matrices are sometimes called $(J,\widetilde{J})$-hyperexchange or $(J,\widetilde{J})$-hypernormal matrices in the signal processing literature [17].

We recall that a tridiagonal matrix is *unreduced* if none of its next-to-diagonal elements (that is, the elements on the first subdiagonal and the first superdiagonal) is zero.

The following result is related to the implicit Q theorem [11]. A more general form can be found in [18, Thm. 2.2].

THEOREM 2.1. *If $C \in \mathbb{R}^{n\times n}$ admits a representation of the form*

$$Q^T C Q = T, \tag{2.1}$$

where T is unreduced tridiagonal and Q is $(J,\widetilde{J})$-orthogonal, then the columns of Q and the next-to-diagonal elements of T are determined up to signs by the first (or last) column of Q.

We give the proof since we need to refer to it later in the text. This is a constructive proof that describes a Lanczos process.

Proof. Let $\widetilde{J} = \operatorname{diag}(\widetilde{\sigma}_i)$, $\widetilde{\sigma}_i = \pm 1$, $i = 1\colon n$, and

$$T = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{bmatrix}.$$

We assume that q_1 is given and normalized such that $\widetilde{\sigma}_1 = q_1^T J q_1$. This yields

$$\alpha_1 = q_1^T C q_1.$$

Using the $(J,\widetilde{J})$-orthogonality of Q, equation (2.1) can be rewritten as

$$JCQ = Q\widetilde{J}T. \tag{2.2}$$

Equating the first column on each side of (2.2) gives

$$p_1 := JCq_1 - \alpha_1\widetilde{\sigma}_1 q_1 = \beta_2\widetilde{\sigma}_2 q_2.$$

From the $(J,\widetilde{J})$-orthogonality of Q we get $\widetilde{\sigma}_2 = \beta_2^{-2} p_1^T J p_1$, which implies

$$\widetilde{\sigma}_2 = \operatorname{sign}(p_1^T J p_1), \quad \beta_2 = \pm\sqrt{|p_1^T J p_1|},$$

so that $q_2 = \widetilde{\sigma}_2\beta_2^{-1} p_1$ is determined up to the sign chosen for β_2. The second diagonal element of T is uniquely determined by

$$\alpha_2 = q_2^T C q_2.$$

Hence, the construction of q_2, α_2, β_2, and $\widetilde{\sigma}_2$ requires just the knowledge of p_1. Now suppose that the first $j < n$ columns of Q and the leading $j \times j$ principal submatrices

of T and $\widetilde{J}$ are known. Then by equating the jth columns on each side of (2.2) we obtain

$$p_j := JCq_j - \widetilde{\sigma}_j\alpha_j q_j - \widetilde{\sigma}_{j-1}\beta_j q_{j-1} = \widetilde{\sigma}_{j+1}\beta_{j+1}q_{j+1}.$$

Using once again the $(J,\widetilde{J})$-orthogonality of Q we have

$$\widetilde{\sigma}_{j+1} = \operatorname{sign}(p_j^T Jp_j), \quad \beta_{j+1} = \pm\sqrt{|p_j^T Jp_j|}. \tag{2.3}$$

Hence

$$q_{j+1} = \widetilde{\sigma}_{j+1}\beta_{j+1}^{-1}p_j, \quad \alpha_{j+1} = q_{j+1}^T Cq_{j+1}. \tag{2.4}$$

Again, β_{j+1} and q_{j+1} are determined up to a sign. By induction on j all columns of Q and all next-to-diagonal elements of T are determined, up to a sign by q_1.

The proof is similar if q_n, the last column of Q is chosen in place of q_1. ∎

For a particular q_1, the proof shows that if, for some $j \le n$, $p_j^T Jp_j = 0$, the reduction breaks down. If $p_j = 0$ then $\beta_{j+1} = 0$. We can carry on the construction with a new q_{j+1} chosen to be J-orthogonal to the previous q_k, $k = 1\colon j$. If $p_j^T Jp_j = 0$ but $p_j \neq 0$ then the breakdown is serious and there is no $(J,\widetilde{J})$-orthogonal matrix Q with this given q_1 that satisfies (2.1). In this case, q_1 is called *exceptional*.

The construction of the quantities q_{j+1}, α_{j+1}, β_{j+1}, and $\widetilde{\sigma}_{j+1}$ in (2.3) and (2.4) corresponds to a modification of the Lanczos process for symmetric matrices and therefore provides a numerical method for the reduction of a symmetric-diagonal pair to tridiagonal-diagonal form. We will instead consider methods based on a finite sequence of unified rotations or unified Householder reflectors or a mix of hyperbolic rotations and Householder reflectors. But before describing the tridiagonalization process we first consider the reduction of the symmetric indefinite pair (A,B) to symmetric-diagonal form.

3. Reduction to symmetric-diagonal form. Since B is indefinite we use a block LDL^T factorization [13, Chap. 11]

$$P^T BP = LDL^T, \tag{3.1}$$

where P is a permutation matrix, L is unit lower triangular and D is diagonal with 1×1 or 2×2 blocks on its diagonal. This factorization costs $n^3/3$ operations plus the cost of determining the permutation matrix. There are several possible choices for P (see [13, sect. 11.1] for a detailed description and stability analysis). We opt for the symmetric rook pivoting strategy [13, sect. 9.1], as it yields a factor L with bounded elements. Let

$$D = X|\Lambda|^{1/2}J|\Lambda|^{1/2}X^T, \quad J \in \operatorname{diag}_q^n(\pm 1), \tag{3.2}$$

be the eigendecomposition of D, where X is orthogonal and Λ is the diagonal matrix of eigenvalues. Note that X has the same structure as D with the 1×1 blocks equal to 1 and the 2×2 blocks can be chosen to be Jacobi rotations of the form

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}, \quad c^2 + s^2 = 1.$$

The pair (C,J) with

$$C = M^T AM, \quad M = PL^{-T}X|\Lambda|^{-1/2} \tag{3.3}$$

is congruent to (A, B) and is in symmetric-diagonal form.

The following pseudocode constructs C, J, and the transformation matrix M in (3.3). We assume that a function computing a LDL^T factorization with rook pivoting is available. For example, we can use the MATLAB function `ldlt_symm` from Higham's Matrix Computation Toolbox [12].

function $[C, J, M] = \text{sym\_diag}(A, B)$
% Compute C, J, and M so that $M^T(A, B)M = (C, J)$
% is a symmetric-diagonal pair.
Compute the factorization $P^T BP = LDL^T$
$X = I$
for $k = 1 : n - 1$
  if $D(k + 1, k) \neq 0$
    $\tau = 0.5(D(k + 1, k + 1) - D(k, k))/D(k + 1, k)$
    if $\tau \geq 0$
      $t = 1/(\tau + \sqrt{1 + \tau^2})$
    else
      $t = -1/(-\tau + \sqrt{1 + \tau^2})$
    end
    $c = 1/\sqrt{1 + t^2}$, $s = tc$
    $X[k\!:\!k + 1, k\!:\!k + 1] = \left[\begin{smallmatrix} c & s \\ -s & c \end{smallmatrix}\right]$
    $\alpha = D(k, k) - D(k + 1, k)t$
    $\beta = D(k + 1, k + 1) + D(k + 1, k)t$
    $D(k\!:\!k + 1, k\!:\!k + 1) = \left[\begin{smallmatrix} \alpha & 0 \\ 0 & \beta \end{smallmatrix}\right]$
  end
end
$J = \text{sign}(D)$,
$C = |D|^{-1/2} X^T L^{-1} (PAP^T) L^{-T} X |D|^{1/2}$
$M = PL^{-T} X |D|^{-1/2}$

We now give a rounding error analysis of this reduction. We use the standard model of floating point arithmetic [13, sect. 2.2]:

$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1 + \delta)^{\pm 1}, \qquad |\delta| \leq u, \quad \operatorname{op} = +, -, *, /,$$

where u is the unit roundoff.

Let $\widehat{L}\widehat{D}\widehat{L}^T$ be the computed factorization in (3.1). Using a general result on the stability of block LDL^T factorization [13, Thm. 11.3], we have

$$P^T(B + \Delta B_1)P = \widehat{L}\widehat{D}\widehat{L}^T, \ |\Delta B_1| \leq p(n)\, u(|B| + P|\widehat{L}||\widehat{D}||\widehat{L}^T|P^T) + O(u^2) \tag{3.4}$$

with p a linear polynomial.

Slapničar [22] shows that when a Jacobi rotation is used to compute the decomposition $H = GJG^T$ of a symmetric $H \in \mathbb{R}^{2\times 2}$ and $J \in \text{diag}(\pm 1)$, the computed decomposition $\widehat{G}\widehat{J}\widehat{G}^T$ satisfies

$$\widehat{G}\widehat{J}\widehat{G}^T = H + \Delta H, \quad |\Delta H| \leq \alpha |G||G^T| u,$$

with α a small integer constant. Using this result we obtain for the computed eigendecomposition (3.2)

$$\widehat{X}|\widehat{\Lambda}|^{1/2}\widehat{J}|\widehat{\Lambda}|^{1/2}\widehat{X}^T = \widehat{D} + \Delta\widehat{D}, \qquad |\Delta\widehat{D}| \leq \tilde{\alpha} u |\widehat{X}|\ |\widehat{\Lambda}|\ |\widehat{X}^T| \tag{3.5}$$

with $\tilde{\alpha}$ a small integer constant. Combining (3.4) with (3.5), we have

$$P^T(B+\Delta B)P = \widehat{L}\widehat{X}\ |\widehat{\Lambda}|^{1/2}\widehat{J}|\widehat{\Lambda}|^{1/2}\widehat{X}^T\widehat{L}^T,$$

where

$$|\Delta B| \le p'(n)u(|B| + P|\widehat{L}|\ |\widehat{X}|\ |\widehat{\Lambda}|\ |\widehat{X}^T|\ |\widehat{L}^T|P^T) + O(u^2).$$

This is the best form of bound we could expect. Note that if rook pivoting is used then all the entries of L are bounded by 2.78 [13, sect. 11.1.3].

Using standard results [13] on the componentwise backward error in solving triangular systems and componentwise backward errors in the product of matrices we find, after some algebraic manipulations, that the computed $\widehat{C}$ satisfies

$$\widehat{C} = |\widehat{\Lambda}|^{-1/2}\widehat{X}^T\widehat{L}^{-1}P^T(A+\Delta A)P\widehat{L}^{-T}\widehat{X}|\widehat{\Lambda}|^{-1/2},$$

where

$$\begin{aligned}|\Delta A| \le \gamma_n\Big(&P|\widehat{L}||\widehat{L}^{-1}||A|(I + |\widehat{L}^{-T}||\widehat{L}^T|P^T)\\ &+ P|\widehat{L}||\widehat{X}||\widehat{X}^T||\widehat{L}^{-1}||A||\widehat{L}^{-T}|(I + |\widehat{X}^T||\widehat{X}|)|\widehat{L}^T|P^T\Big)\end{aligned}$$

with $\gamma_n = nu/(1-nu)$. Taking the ∞-norm gives

$$\|\Delta A\|_\infty \le \widetilde{\gamma}_n\kappa_\infty(L)^2\|A\|_\infty, \tag{3.6}$$

with $\widetilde{\gamma}_n = cnu/(1-cnu)$, c being a small integer constant.

This is the same form of normwise backward error result as we obtain for the reduction of a symmetric definite pair (A,B) with B positive definite using a Cholesky decomposition of B [7]. If rook pivoting is used in the block LDL^T factorization then [13, Prob. 8.5]

$$\kappa_\infty(L) = \|L\|_\infty\|L^{-1}\|_\infty \le 3.78^{n-1}\big(1 + 2.78(n-1)\big),$$

and so $\|\Delta A\|_\infty$ in (3.6) is bounded independently of B. For the definite case, if complete pivoting in the Cholesky factorization is used, we have the smaller bound $\kappa_\infty(L) \le n2^{n-1}$.

4. Reduction to tridiagonal-diagonal form. Given a symmetric-diagonal pair (C,J) with $J \in \operatorname{diag}_q^n(\pm 1)$, this section deals with the construction of a nonsingular matrix Q such that

$$Q^TCQ = T, \qquad Q^TJQ = \widetilde{J}, \tag{4.1}$$

with T symmetric tridiagonal and $\widetilde{J} \in \operatorname{diag}_n^q(\pm 1)$. We denote by σ_i and $\widetilde{\sigma}_i$ the ith diagonal element of J and $\widetilde{J}$, respectively.

Brebner and Grad [5] propose two methods: a pseudosymmetric Givens method and a pseudosymmetric Householder method. Both reduce the pseudosymmetric[1] matrix JC to pseudosymmetric tridiagonal form $\widetilde{T} = \widetilde{J}T$ with $\widetilde{J} \in \operatorname{diag}_q^n(\pm 1)$ and T symmetric tridiagonal. Their reduction is equivalent to reducing $C - \lambda J$ to symmetric

[1] A matrix M is pseudosymmetric if $M = NJ$ where $N = N^T$ and $J = \operatorname{diag}(\pm 1)$. Equivalently, MJ (or JM) is symmetric.

tridiagonal-diagonal form $T - \lambda\widetilde{J}$ using a sequence of Givens and hyperbolic transformations or a sequence of hyperbolic Householder transformations. The first two reductions described below are based on similar ideas. They contain several improvements over Brebner and Grad's reductions that make them more stable. The third reduction is new and based on a combination of Householder reflectors and hyperbolic rotations.

4.1. Reduction by unified rotation. The term *unified rotation* was introduced by Bojanczyk, Qiao, and Steinhardt [4]. Unified rotations include both orthogonal and hyperbolic rotations. Given a 2×2 signature matrix $J = \operatorname{diag}(\sigma_1, \sigma_2)$, unified rotations have the form

$$G = \begin{bmatrix} c & \frac{\sigma_2}{\sigma_1}s \\ -s & c \end{bmatrix} \in \mathbb{R}^{2\times 2}, \quad \sigma_1 c^2 + \sigma_2 s^2 = \widetilde{\sigma}_1, \quad \widetilde{\sigma}_1 = \pm 1. \tag{4.2}$$

If we define $\widetilde{\sigma}_2 = \sigma_2\widetilde{\sigma}_1/\sigma_1$ then $G^T J G = \operatorname{diag}(\widetilde{\sigma}_1, \widetilde{\sigma}_2) \equiv \widetilde{J}$, that is, G is $(J, \widetilde{J})$-orthogonal. Thus G is a Givens rotation when $J = \pm I$ and a hyperbolic rotation when $J \neq \pm I$. Hyperbolic rotations are said to be of *type* 1 when $J = \widetilde{J}$ and of *type* 2 when $J = -\widetilde{J}$. Let $x = [x_1, x_2]^T \neq 0$ be such that $x^T J x \neq 0$. Choosing

$$c = x_1/\sqrt{|x^T J x|}, \qquad s = x_2/\sqrt{|x^T J x|}$$

gives $Gx = [\rho, 0]^T$ with $\rho = (\widetilde{\sigma}_1/\sigma_1)\sqrt{|x^T J x|}$ and $\widetilde{\sigma}_1 = \operatorname{sign}(x^T J x)$.

The following pseudocode, inspired by [4, Alg. 2] constructs c and s and guards against the risk of overflow.

```
function [c, s, J̃] = u_rotate(x, J)
% Given x = [x1, x2]^T and J = diag(σ1, σ2), compute c and s defining the
% unified rotation G such that Gx has zero second element and G is
% (J, J̃)-orthogonal.
γ = σ2/σ1, J̃ = J
if x2 = 0
   s = 0, c = 1, return
end
if |x1| = −γ|x2|
   No unified rotation exists—abort.
end
if |x1| > |x2|
   t = x2/x1, τ = 1 + γ t^2
   c = sign(x1)/√τ, s = ct
else
   t = x1/x2, τ = γ + t^2
   s = sign(x2)/√|τ|, c = st
end
if τ < 0, J̃ = −J̃, end
```

Bojanczyk, Brent, and Van Dooren [2] noticed that how hyperbolic rotations are applied to a vector is crucial to the stability of the computation. Consider the computation of $y = Gx$ with $\sigma_2/\sigma_1 = -1$:

$$y_1 = cx_1 - sx_2, \tag{4.3}$$

$$y_2 = -sx_1 + cx_2. \tag{4.4}$$

We call (4.3)–(4.4) the *direct application* of G to a vector x. When $\sigma_1 = \widetilde{\sigma}_1$ (i.e., for hyperbolic rotations of type 1), we compute y_1 from (4.3). Solving (4.3) for x_1 gives

$$x_1 = \frac{y_1}{c} + \frac{s}{c}x_2, \tag{4.5}$$

which allows (4.4) to be rewritten as

$$y_2 = -\frac{s}{c}y_1 + \left(-\frac{s^2}{c} + c\right)x_2 = -\frac{s}{c}y_1 + \frac{x_2}{c}. \tag{4.6}$$

Note that (4.5) and (4.6) can be rewritten as

$$\begin{bmatrix} x_1 \\ y_2 \end{bmatrix} = \widetilde{G}\begin{bmatrix} y_1 \\ x_2 \end{bmatrix}, \quad \widetilde{G} = \begin{bmatrix} 1/c & s/c \\ -s/c & 1/c \end{bmatrix},$$

and $\widetilde{G}$ is an orthogonal Givens rotation. As multiplication of a vector by a Givens rotation is a stable process, this suggests that the computation of y_2 is likely to be more stable using (4.6) than using (4.4). We call (4.3), (4.6) the *mixed application* of G to a vector x. Similar formulas can be derived for hyperbolic rotations of type 2. Finally we note that the two matrices G and $\widetilde{G}$ are related by the *exchange operator*, $G = \mathrm{exc}(\widetilde{G})$. The exchange operator has a number of interesting mathematical properties; see Higham [14]. In particular, it maps J-orthogonal matrices to orthogonal matrices and vice-versa.

We express the application of unified rotations as follows.

```
function B = r_apply(c, s, J, J̃, B)
% Apply hyperbolic rotation defined by c, s, J, and J̃ to 2 × n matrix B.
γ = J(2,2)/J(1,1), σ̃1 = J̃(1,1)
for j = 1:n
    x = B(1,j)
    B(1,j) = cB(1,j) + γsB(2,j)
    if γ = 1
            B(2,j) = −sx + cB(2,j) % Givens rotation
    elseif σ1 = σ̃1
            B(2,j) = −(s/c)B(1,j) + B(2,j)/c % Rotation of type 1
    else
            B(2,j) = −(c/s)B(1,j) − x/s % Rotation of type 2
    end
end
```

The importance of applying hyperbolic rotations to a vector or a matrix in a mixed way is illustrated in section 6.1.

Unified rotations can be used for reducing $C - \lambda J$ to tridiagonal-diagonal form in a way similar to how Givens rotations are used to tridiagonalize a symmetric matrix (Givens method) [10], [19]. Assume that at the beginning of step j the matrix $C = (c_{ij})$ is tridiagonal as far as its first $j-1$ rows and columns are concerned. At the jth step, we introduce zeros in the matrix C in positions (i,j) and (j,i), $j+2 \le i \le n$ using $n-j-1$ unified rotations. The zeroing operations can be done, for example, in the natural order $j+2, j+3, \ldots, n$ or the reverse order. The element in position (i,j) is annihilated by a unified rotation in the plane (k,i), where k is chosen so that $k < j$, $k \ne i$ and $c_{kj} \ne 0$. The signature matrix is modified each time a hyperbolic rotation of type 2 is applied. The matrix Q which accumulates the product of all the

unified rotations satisfies

$$Q^T C Q = T, \qquad Q^T J Q = \widetilde{J} \in \operatorname{diag}_q^n(\pm 1).$$

The number of rotations required is of order $n^2/2$. The reduction fails if at some stage $\sigma_i |c_{ij}| = \sigma_k |c_{kj}| \neq 0$, where σ_j denotes the jth diagonal elements of J.

For the standard case ($J = I$), the most popular choices for the rotation plane (k, i) are $k = j + 1$ or $k = i - 1$, either choice yielding a perfectly stable reduction. However, when $J \neq I$, the choice of k is crucial for the stability of the reduction. Indeed, using a result of Ostrowski [15, p. 224] one can show that inherent relative errors in a symmetric matrix A can be magnified by as much as $\kappa(Q)^2$ in passing to $Q^T A Q$ for any nonsingular Q [8]. Clearly, $\kappa(G) = 1$ for Givens rotations, but for hyperbolic rotations [4]

$$\kappa(G) = \frac{|c| + |s|}{\big||c| - |s|\big|}, \tag{4.7}$$

which can be arbitrarily large. Hence it is advisable to use as few hyperbolic rotations as possible.

Recall that at stage j of the reduction we need to zero all the elements in rows $j + 2$ up to n of the jth column. First, we perform all possible Givens rotations in planes (ℓ, i) with $\sigma_\ell = \sigma_i$, $j + 1 \leq \ell < i \leq n$. At this point, either the stage is finished or there are two nonzero entries left in positions $(j + 1, j)$ and (i, j) with i such that $j+1 < i \leq n$ and $\sigma_{j+1} = -\sigma_i$. Then a single hyperbolic rotation in the plane $(j+1, i)$ does the final elimination. This strategy has two main advantages. First, it reduces the number of hyperbolic rotations used during the reduction process to at most $n-2$. Secondly, it minimizes the risk of having two hyperbolic rotations acting in the same plane. This tends to reduce the growth of rounding errors and increases the chance that the largest condition number of the individual transformations will be of the same order of magnitude as the condition number of the overall transformation Q. The complete algorithm is summarized as follows.

ALGORITHM 4.1 (tridiagonalization by unified rotations). *Given an $n \times n$ symmetric matrix C and a signature matrix $J \in \operatorname{diag}_q^n(\pm 1)$, the following algorithm overwrites C with the tridiagonal matrix $T = Q^T C Q$ and J with $Q^T J Q \in \operatorname{diag}_q^n(\pm 1)$, Q being the product of unified rotations.*

```
for j = 1: n − 2
    i_h = 0, i = n
    while i > j + 1 or i_h > 0
        if i > j + 1
            Find largest k, j + 1 ≤ k ≤ i, such that J_ii = J_kk.
            rot = [k i]
        else
            rot = [j + 1 i_h], i_h = 0
        end
        if rot(1) = rot(2)
            i_h = rot(1)
        else
            [c, s, J_temp] = u_rotate(C(rot, j), J(rot, rot))
            C(rot, j: n) = r_apply(c, s, J(rot, rot), J_temp, C(rot, j: n))
            C(j: n, rot) = r_apply(c, s, J(rot), J_temp, C(j: n, rot))^T
            C(i, j) = 0; C(j, i) = 0, J(rot, rot) = J_temp
```

```
            end
            i = i - 1
        end
    end
```

The major differences between Algorithm 4.1 and Brebner and Grad's pseudosymmetric Givens algorithm [5] are that in the latter algorithm there is no particular strategy to minimize the number of hyperbolic rotations used and the hyperbolic rotations are applied directly to CJ (instead of as in function r_apply above).

4.2. Reduction by unified Householder reflectors. Unified Householder reflectors [4] include standard orthogonal Householder transformations [11] together with hyperbolic Householder reflectors [20], [21]. Given a signature matrix $J = \mathrm{diag}(\sigma_i)$, a unified Householder matrix has the form

$$H = H(J,k,v) = P\Big(J - \frac{2vv^T}{v^TJv}\Big), \quad v^TJv \neq 0, \tag{4.8}$$

where P is a permutation matrix in the $(1,k)$-plane.

For any vector x such that $x^TJx \neq 0$, the unified Householder vector v can be chosen so that H maps x onto the first column of the identity matrix. Let k be such that $e_k^TJe_k = \sigma_k := \mathrm{sign}(x^TJx)$ and let

$$v = Jx + \sigma_k \mathrm{sign}(x_k)|x^TJx|^{1/2}e_k. \tag{4.9}$$

Then it is easy to check that $v^TJv \neq 0$ and that $Hx = -\sigma_k\mathrm{sign}(x_k)|x^TJx|^{1/2}e_1$. Note also that P^TH is J-orthogonal.

The application of a hyperbolic Householder matrix to a vector can be done either directly, as

$$Hx = P\left(Jx - \frac{2v^Tx}{v^TJv}v\right),$$

or, as for hyperbolic rotations, in a mixed way making use of the orthogonal matrix $\mathrm{exc}(H)$. Stewart and Stewart [23] show that both approaches are mixed-forward backward stable. We use the first approach since it yields simpler coding.

In [4] it is shown that

$$\sigma_{\min}^{-1}(H) = \sigma_{\max}(H) = \frac{v^Tv}{|v^TJv|} + \sqrt{\left(\frac{v^Tv}{v^TJv}\right)^2 - 1}. \tag{4.10}$$

For $J = I$, $\sigma_{\min} = \sigma_{\max} = 1$ and for $J \neq I$ the ratio v^Tv/v^TJv can be arbitrarily large. Fortunately, there is some freedom in the choice of the plane $(1,k)$ for the permutation P. Choosing k so that

$$e_k^TJe_k = \mathrm{sign}(x^TJx) \quad \text{and} \quad |x_k| \text{ is maximized}$$

minimizes the ratio v^Tv/v^TJv and therefore minimizes $\kappa(H)$. This is the pivoting strategy proposed in [4], [23].

The following pseudocode inspired by [4, Alg. 3] determines the permutation matrix P and constructs the unified Householder vector.

```
function [v, k, β, α] = u_house(x, J)
% Determine the permutation P in the (1, k) plane and compute v, α, and β
```

% such that $H = P(J - \beta vv^T)$ satisfies $Hx = -\alpha e_1$ with $P^T H$ J-orthogonal.
if $x^T Jx = 0$
    No hyperbolic Householder exists—abort.
end
$m = \|x\|_\infty$, $x = x/m$
if $J = \pm I$
    $k = 1$
else
    Find k so that $|x_k|$ is maximized and $\text{sign}(x_k^T Jx_k) = J_{kk}$.
end
$\alpha = J_{kk}\text{sign}(x_k)|x^T Jx|^{1/2}$
$v = Jx + \alpha e_k$
$\beta = 2/(v^T Jv)$
$\alpha = m\alpha$

The symmetric matrix C can be reduced to tridiagonal form while keeping the diagonal form of J by $n-2$ unified Householder transformations. Each transformation annihilates the required part of a whole column and whole corresponding row. The complete algorithm is summarized below.

ALGORITHM 4.2 (tridiagonalization by unified Householder reflectors). *Given an $n \times n$ symmetric matrix C and a signature matrix $J \in \text{diag}_q^n(\pm 1)$, the following algorithm overwrites C with the tridiagonal matrix $T = Q^T CQ$ and J with $Q^T JQ \in \text{diag}_q^n(\pm 1)$, Q being the product of unified Householder reflectors.*

for $j = 1\colon n-2$
    $ind = j+1\colon n$
    $[v, k, \beta, \alpha] = \text{u\_house}(C(ind, j), J(ind, ind))$
    Swap rows and columns $j+1$ and $j+k$ of C.
    $C(ind, j) = -\alpha e_1$, $C(j, ind) = C(ind, j)^T$
    $p = \beta J(ind, ind)C(ind, ind)v$
    $w = p - \beta^2(v^T C(ind, ind)v)v/2$
    $C(ind, ind) = J(ind, ind)C(ind, ind)J(ind, ind) - wv^T - vw^T$
end

Note that the reduction fails if at some stage j, $x^T Jx = 0$, where $x = C(j+1\colon n, j)$.

Algorithm 4.2 differs from the pseudosymmetric Householder algorithm in [5] in that, for the latter algorithm, Brebner and Grad use a rank-one update H of the form $H = I - 2Jvv^T$, where v can have complex entries even though H is real. This vector is not computed but, instead, the transformation H is computed element by element and applied explicitly to CJ, which is a costly operation. Also, no pivoting is used to reduce the condition number of the transformations.

4.3. Reduction by a mix of Householder reflectors and hyperbolic rotations. Here we adapt an idea developed by Bojanczyk, Higham, and Patel [3] for hyperbolic QR factorizations of rectangular matrices. We propose a tridiagonalization that uses a combination of Householder reflectors and hyperbolic rotations. As hyperbolic rotations are not norm-preserving, we aim to use a minimal number of them.

Assume for notational simplicity that $J = \text{diag}(I_p, -I_q) \in \text{diag}_q^n(\pm 1)$ and partition $x \in \mathbb{R}^n$ so that $x_p = x(1\colon p)$ and $x_q = x(p+1\colon n)$. We first define a $(J, \widetilde{J})$-orthogonal matrix that maps x into the first column of the identity matrix. Let H_p and H_q be two Householder matrices defined so that

$$H_p x_p = -\|x_p\| e_1, \qquad H_q x_q = -\|x_q\| e_1.$$

Next we define a 2×2 hyperbolic rotation such that

$$\begin{bmatrix} c & -s \\ -s & c \end{bmatrix} \begin{bmatrix} \|x_p\| \\ \|x_q\| \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}, \quad \alpha \in \mathbb{R},$$

and build from it an $n \times n$ hyperbolic rotation G in the $(1, p+1)$ plane. Then the matrix

$$S = G \begin{bmatrix} H_p & 0 \\ 0 & H_q \end{bmatrix} \tag{4.11}$$

maps x into the first column of the identity matrix and satisfies $\widetilde{J} \equiv S^T J S \in \operatorname{diag}_q^n(\pm 1)$. Note that $\widetilde{J} = J$ when G is a hyperbolic rotation of type 1 and if G is a hyperbolic rotation of type 2 then $\widetilde{J}$ and J are identical except in position $(1,1)$ and $(p+1, p+1)$, where their signs differ. From (4.11) and (4.7), the condition number of S is given by

$$\kappa(S) = \frac{\|x_p\| + \|x_q\|}{\big|\|x_p\| - \|x_q\|\big|}. \tag{4.12}$$

Unlike for the tridiagonalization via unified Householder matrices, we have no free parameters that can be used to minimize $\kappa(S)$. The next result shows that $\kappa(S)$ is already of optimal condition relative to unified Householder matrices.

THEOREM 4.3. *Let H be a unified Householder reflector as in* (4.8) *and let S be a combination of Householder reflectors and a hyperbolic rotation as in* (4.11)*, both mapping a vector x to a multiple of e_1, the first column of the identity matrix. Then*

$$\kappa(S) \le \kappa(H).$$

If R denotes the matrix which accumulates the product of the Givens rotations and the hyperbolic rotation mapping x to a multiple of e_1 as described in section 4.1*, then*

$$\kappa(R) = \kappa(S).$$

Proof. Let $H = P(J - \beta v v^T)$, where

$$\beta = 2/v^T J v, \quad v = Jx + \sigma_k \operatorname{sign}(x_k) |x^T J x|^{1/2} e_k \tag{4.13}$$

for some k such that $e_k^T J e_k = \sigma_k = \operatorname{sign}(x^T J x)$ and P is a permutation in the $(1,k)$-plane. Assume that $J = (I_p, -I_q)$ and partition v and x accordingly:

$$\begin{aligned} v_p &= v(1\colon p), & x_p &= x(1\colon p), \\ v_q &= v(p+1\colon p+q), & x_q &= x(p+1\colon p+q). \end{aligned}$$

From (4.10),

$$\kappa(H) = \frac{\sigma_{\max}(H)}{\sigma_{\min}(H)} = \left(\frac{v^T v + \sqrt{(v^T v)^2 - (v^T J v)^2}}{v^T J v} \right)^2 = \left(\frac{\|v_p\| + \|v_q\|}{\|v_p\| - \|v_q\|} \right)^2.$$

Suppose that $\kappa(H) < \kappa(S)$. There is no loss of generality in assuming that $\|x_p\| > \|x_q\|$. Using the expression for $\kappa(S)$ in (4.12) we have

$$(\|x_p\| + \|x_q\|)(\|v_p\| - \|v_q\|)^2 > (\|x_p\| - \|x_q\|)(\|v_p\| + \|v_q\|)^2,$$

or equivalently,

$$2\|x_p\|\|v_p\|\|v_q\| - \|x_q\|\|v_p\|^2 - \|x_q\|\|v_q\|^2 < 0. \tag{4.14}$$

Since $\|x_p\| > \|x_q\|$, $\operatorname{sign}(x^T Jx) > 0$ and hence $1 \le k < p$. From (4.13), $\|v_q\| = \|x_q\|$ and

$$\|v_p\|^2 = 2\|x_p\|^2 - \|x_q\|^2 + 2|x_k|(\|x_p\|^2 - \|x_q\|^2)^{1/2} = \|x_p\|^2(2\mu - \alpha),$$

where $\mu = 1 + |x_k|(1-\alpha)^{1/2}/\|x_p\|$ and $\alpha = \|x_q\|^2/\|x_p\|^2$. Using these expressions for $\|v_p\|$ and $\|v_q\|$ in inequality (4.14) leads to

$$f(\mu) = 3\mu^2 - 4\mu\alpha + \alpha^2 < 0,$$

which is satisfied for $\frac{\alpha}{3} < \mu < \alpha < 1$. But by the definition of μ, $1 \le \mu \le 1+(1-\alpha)^{1/2}$ and for these values of μ, $f(\mu) \ge 0$. Hence $\kappa(H) \ge \kappa(S)$.

The equality $\kappa(R) = \kappa(S)$ is obvious. □

Assume that (C, J) has been permuted so that $J = \operatorname{diag}(I_p, -I_q)$. Again, as in the previous section, we can transform C to tridiagonal form while preserving the diagonal form of J by $n-2$ transformations of the form (4.11). The key point in the reduction is that at each step the part of the signature matrix involved in the transformation is of the form $\operatorname{diag}(I_{\tilde{p}_j}, -I_{\tilde{q}_j})$, $\tilde{p}_j + \tilde{q}_j = n - j$. Note that if we reach the stage where $\tilde{p}_j = 0$ or $\tilde{q}_j = 0$ then the rest of the reduction is carried out with orthogonal Householder matrices only.

This reduction uses at least $\min(p-1, q)$ hyperbolic rotations and at most $n-2$. The smallest number $\min(p-1, q)$ occurs when all the transformations in the reduction process are derived from hyperbolic rotations of type 1. The largest number, $n-2$, happens if hyperbolic rotations of type 2 are used at each step of the reduction. Note that the reduction fails if at some stage j, $\|x_{\tilde{p}_j}\| = \|x_{\tilde{q}_j}\| \ne 0$.

ALGORITHM 4.4 (tridiagonalization by mixed Householder reflectors-hyperbolic rotations). *Given an $n \times n$ symmetric matrix C and a signature matrix $J = (I_p, -I_q)$, the following algorithm overwrites C with the tridiagonal matrix $T = Q^T CQ$ and J with $Q^T JQ \in \operatorname{diag}_q^n(\pm 1)$, Q being the product of mixed Householder reflectors–hyperbolic rotations.*

```
for j = 1:n-2
    p = max(0, p-1)
    if p > 1
        ind = j+1:j+p
        [v, k, β, α] = u_house(C(j+1:j+p, j), J(ind, ind))
        C(ind, j:n) = C(ind, j:n) − βv(v^T C(ind, j:n))
        C(j:n, ind) = C(j:n, ind) − β(C(j:n, ind)v)v^T
    end
    if q > 1
        ind = (j+p+1:n)
        [v, k, β, α] = u_house(C(ind, j), J(ind, ind))
        C(ind, j:n) = C(ind, j:n) − βv(v^T C(ind, j:n))
        C(j:n, ind) = C(j:n, ind) − β(C(j:n, ind)v)v^T
    end
    if p > 0 and q > 0
        rot = [j+1, j+p+1]
        [c, s, J_temp] = u_rotate(C(rot, j), J(rot, rot))
```

```
            C(rot, j:n) = r_apply(c, s, J(rot, rot), J_temp, C(rot, j:n))
            C(j:n, rot) = r_apply(c, s, J(rot), J_temp, C(j:n, rot)^T))^T
            C(i, j) = 0; C(j, i) = 0
            if J(rot, rot) = -J_temp
                  p = p + 1, q = q - 1
                  J(rot, rot) = J_temp
            end
        end
    end
```

We cannot conclude from Theorem 4.3 that Algorithms 4.1 and 4.4 are more stable than Algorithm 4.2 since at step k of the tridiagonalization process the column of C to be annihilated is not the same for each reduction. However, intuitively, we may expect Algorithms 4.1 and 4.4 to behave better than Algorithm 4.2.

5. Monitoring condition numbers and preventing breakdown. If serious breakdown occurs during the reduction to tridiagonal-diagonal form (see the end of section 2) then we can permute C and start again. This is equivalent to restarting the Lanczos process described in the proof of Theorem 2.1 with a new vector q_1. Of course, the major disadvantage with this approach is that all the previous computation is lost. We take an alternative approach, based on an idea from Geist, Lu, and Wachpress [9] for curing breakdown occurring in the tridiagonalization of nonsymmetric matrices.

If breakdown occurs at step j of the reduction process or if the condition number of the next transformation is too large, we apply a unified rotation $\widetilde{G}$ on the two first rows and columns of the current C. This brings nonzero values in positions $(3,1)$ and $(1,3)$. This bulge in the tridiagonal form is chased down the matrix from position $(3,1)$ to $(4,2)$ and so on via $j-2$ unified rotations. This chasing procedure costs $O(j)$ operations and the result is a new column j in C. The whole procedure may be tried again if some large condition numbers occurs before the reduction is completed.

In our implementation the unified rotation $\widetilde{G}$ is generated randomly but with the constraint that $\kappa(\widetilde{G}) = O(1)$.

6. Numerical experiments. Our aim in this section is to investigate the numerical properties of the tridiagonal-diagonal reduction algorithms just described. We name our MATLAB implementations

- `trd_ur`: tridiagonalization by unified rotations (Algorithm 4.1),
- `trd_uh`: tridiagonalization by unified Householder reflectors (Algorithm 4.2),
- `trd_hr`: tridiagonalization by mixed Householder reflectors-hyperbolic rotations (Algorithm 4.4).

Given a symmetric matrix C and a signature matrix J we formed explicitly, during the course of the reduction, the transformation Q such that $T = Q^T C Q$ is tridiagonal and $\widetilde{J} = Q^T J Q$ is a signature matrix. The following quantities were computed:

- the scaled residual error and departure from $(J, \widetilde{J})$-orthogonality

$$\mathcal{R} = \frac{\|Q^T C Q - T\|}{\|C\|\|Q\|^2}, \qquad \mathcal{O} = \frac{\|Q^T J Q - \widetilde{J}\|}{\|Q\|^2}, \tag{6.1}$$

- $\kappa(Q)$, the condition number of the transformation Q,
- the largest condition numbers,

$$\kappa_G = \max_k \kappa(G_k), \quad \kappa_H = \max_k \kappa(H_k), \quad \kappa_S = \max_k \kappa(S_k),$$

of the transformations used to zero parts of the matrix C. Here G, H, and S

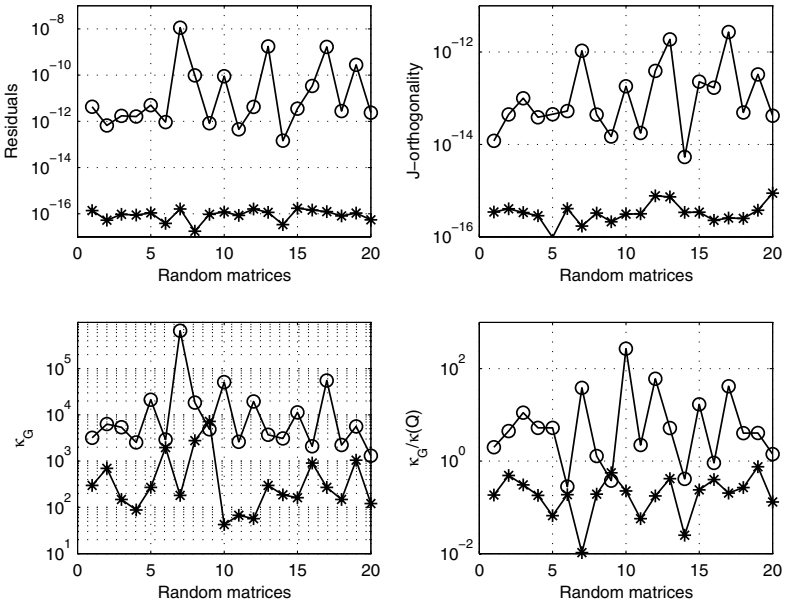

FIG. 6.1. *Residuals and condition numbers for* 20 *random matrices. Results from* `trd_BG1` *are marked with* "o" *and results from* `trd_ur` *are marked with* "$*$."

refer to unified rotation, unified Householder reflector, and a combination of two Householder reflectors and one hyperbolic rotation, respectively.

6.1. Tridiagonalization by unified rotations. We first compare `trd_ur` to an implementation of Brebner and Grad's pseudosymmetric Givens method named `trd_BG1`. We ran a set of tests with matrices of the form

```
C = randn(n); C = C+C'; J = mysign(randn(n));
```

where `mysign` is a sign function defined so that `mysign`$(0) = 1$. The residual $\mathcal{R}$ and the departure from $(J, \widetilde{J})$-orthogonality $\mathcal{O}$ as defined in (6.1) are plotted on the top left and right in Figure 6.1 for twenty random matrices of size $n = 50$. Results obtained by `trd_BG1` are plotted with "o" and we use "$*$" for results from `trd_ur`. On this set of matrices, the residuals $\mathcal{R}$ and $\mathcal{O}$ from `trd_ur` are smaller than the ones from `trd_BG1` by a factor as large as 10^7 for $\mathcal{R}$ and 10^4 for $\mathcal{O}$. For a given test problem (C, J), `trd_BG1` and `trd_ur` both compute the same Q, but the construction of Q differs since it is obtained by a different sequence of transformations. The left-hand plot at the bottom of Figure 6.1 helps to compare the largest condition numbers κ_G of the individual transformations used by each algorithm during the reduction process. It shows that κ_G is nearly always smaller for `trd_ur`. Not surprisingly, large values of κ_G correspond to test problems with large values of $\mathcal{R}$ and $\mathcal{O}$. The right-hand plot at the bottom of Figure 6.1 compares both algorithms' ratios $\kappa_G/\kappa(Q)$. Interestingly, for `trd_ur`, κ_G is always smaller than the condition number of the overall transformation Q whereas κ_G is in general larger than $\kappa(Q)$ for `trd_BG1`. The four plots on Figure 6.1 illustrate the numerical superiority of our tridiagonalization using unified rotations over Brebner and Grad's pseudosymmetric Givens method. The improvements are due to the way we apply the rotations and our zeroing strategy.

TABLE 6.1
Comparison between explicit and implicit application of hyperbolic rotations to matrices.

$\mathcal{R}_d$	$\mathcal{R}_m$	$\kappa(Q)$	κ_G	$\mathcal{E}_d$	$\mathcal{E}_m$	$\mathrm{cond}(\lambda)$
2×10^{-12}	2×10^{-15}	3.02	2×10^{3}	4×10^{-10}	2×10^{-13}	4×10^{2}

To emphasize the fact that how hyperbolic rotations are applied to a matrix may be crucial to the stability of the computation we use the direct search maximization routine `mdsmax` of the MATLAB Matrix Computation Toolbox [12] to maximize both ratios $\mathcal{R}_d/\mathcal{R}_m$ and $\mathcal{R}_m/\mathcal{R}_d$. The subscripts d and m stand for direct and mixed, respectively, depending on how the hyperbolic rotations are applied to C during the course of the reduction. We used `trd_BG1` with an option on how to apply the rotations. We found that for some matrix pairs (C, J), $\mathcal{R}_d \gg \mathcal{R}_m$ but when $\mathcal{R}_m$ is larger than $\mathcal{R}_d$, $\mathcal{R}_m \lesssim \mathcal{R}_d$ always. Table 6.1 provides some relevant quantities for a 5×5 pair (C, J) generated by `mdsmax`. We also compared the eigenvalues λ_i of the initial pair (C, J) with those $\tilde{\lambda}_i$ of $(T, \widetilde{J})$ and their corresponding relative condition numbers $\mathrm{cond}(\lambda_i)$,

$$\mathrm{cond}(\lambda_i) = \frac{\|x_i\|\|y_i\|}{|\lambda_i|\ |y_i^* J x_i|},$$

where x_i and y_i are the corresponding right and left eigenvectors. We denote by

$$\mathcal{E} = \max_{i=1:n} \frac{|\lambda_i - \tilde{\lambda}_i|}{|\lambda_i|}$$

the largest relative error for the computed eigenvalues. For this particular example, $\mathcal{R}_d \approx 10^3 \mathcal{R}_m$. Since $\kappa(Q) = O(1)$, it is reasonable to expect $\mathcal{R} = O(u)$ which is clearly not the case when direct application of unified rotations is used. The table also shows that a large value for the residual $\mathcal{R}_d$ directly affects the accuracy to which the eigenvalues are computed from $(T, \widetilde{J})$.

6.2. Tridiagonalization by unified Householder reflectors. We now compare `trd_uh` to an implementation of Brebner and Grad's pseudosymmetric Householder method named `trd_BG2`. The main numerical difference between the two algorithms is that `trd_uh` uses a pivoting strategy aimed to reduce the condition numbers of the unified Householder reflectors. We ran a sequence of tests similar to the ones described in section 6.1. Results are plotted in Figure 6.2 for twenty random test problems of dimension 50. These plots clearly illustrate that the pivoting strategy helps to reduce the residuals and the departure from $(J, \widetilde{J})$-orthogonality. For this set of examples, $\mathcal{R}$ and $\mathcal{O}$ are reduced on average by a factor 10^2 and 10, respectively; the reduction factor is as large as 10^3 for $\mathcal{R}$ and as large as 10^2 for $\mathcal{O}$. As expected, κ_H for `trd_uh` is always smaller than κ_H for `trd_BG2` by a factor as large as 10^3. Recall that small κ_H are essential for the stability of the reduction.

6.3. Comparison of the three reductions. For a particular symmetric-diagonal pair (C, J) with $J = \mathrm{diag}(I_p, -I_q)$ we know that, from Theorem 2.1, the three algorithms produce up to signs the same matrix Q and tridiagonal matrix T. They differ numerically in the way Q is formed.

We generated a large set of problems with matrices C of the form

```
C = randn(n); C = C+C'
```

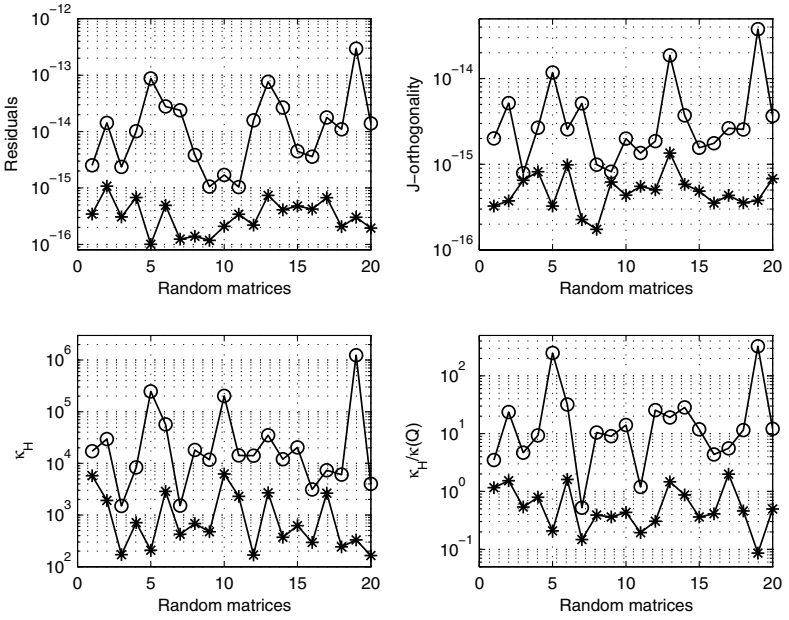

FIG. 6.2. *Residuals and condition numbers for* 20 *random matrices. Results from* `trd_BG2` *are marked with* "o" *and results from* `trd_uh` *are marked with* "$*$."

and

```
C = gallery('randsvd',n); C = C+C'
```

and also matrices $C = Q^T T Q$ obtained from random tridiagonal matrices T and random J-orthogonal matrices Q with prescribed condition numbers. Higham's algorithm [14] was used to generate the random Q.

We ran extensive tests with these types of problems. Here is a summary of our findings.

- As expected, `trd_ur`, `trd_uh` yield residuals of the same order of magnitude.
- 80% of the time, `trd_uh` has residuals of the same order of magnitude as `trd_hr` or `trd_ur`.
- In 20% of the cases where the residuals have different orders of magnitude, `trd_uh` appears the least stable. On average, the residuals and departure from $(J, \widetilde{J})$-orthogonality are 10 times larger with `trd_uh` than with `trd_ur` or `trd_hr`.
- Most of the time, κ_G and κ_S are smaller than κ_H, which is consistent with the previous bullet. Large condition numbers for the individual transformations directly affect the residuals.
- When $\kappa(Q)$ is large the $(J, \widetilde{J})$-departure from orthogonality of Q tends to be larger with `trd_uh` than with the two others algorithms.

This battery of tests seems to indicate that amongst the three reductions `trd_uh` is the least stable. Since `trd_ur` is nearly twice more costly than `trd_hr`, we suggest to use the latter, that is, to use a combination of Householder reflectors and hyperbolic rotations (Algorithm 4.4) to reduce a symmetric-diagonal pair to tridiagonal-diagonal form. We would like to emphasize that in most instances the three algorithms all produce residuals close to what we would expect from a stable algorithm.

Acknowledgment. I thank the referees for valuable suggestions that improved the paper.

REFERENCES

[1] D. A. Bini, L. Gemignani, and F. Tisseur, *The Ehrlich-Aberth Method for the Nonsymmetric Tridiagonal Eigenvalue Problem*, Numerical Analysis Report 428, Manchester Centre for Computational Mathematics, Manchester, England, 2003.

[2] A. Bojanczyk, R. P. Brent, P. Van Dooren, and F. R. de Hoog, *A note on downdating the Cholesky factorization*, SIAM J. Sci. Statist. Comput., 8 (1987), pp. 210–221.

[3] A. Bojanczyk, N. J. Higham, and H. Patel, *Solving the indefinite least squares problem by hyperbolic QR factorization*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 914–931.

[4] A. W. Bojanczyk, S. Qiao, and A. O. Steinhardt, *Unifying unitary and hyperbolic transformations*, Linear Algebra Appl., 316 (2000), pp. 183–197.

[5] M. A. Brebner and J. Grad, *Eigenvalues of $Ax = \lambda Bx$ for real symmetric matrices A and B computed by reduction to a pseudosymmetric form and the HR process*, Linear Algebra Appl., 43 (1982), pp. 99–118.

[6] A. Bunse-Gerstner, *An analysis of the HR algorithm for computing the eigenvalues of a matrix*, Linear Algebra Appl., 35 (1981), pp. 155–173.

[7] P. I. Davies, N. J. Higham, and F. Tisseur, *Analysis of the Cholesky method with iterative refinement for solving the symmetric definite generalized eigenproblem*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 472–493.

[8] S. D. Garvey, F. Tisseur, M. I. Friswell, and J. E. T. Penny, *Simultaneous tridiagonalization of two symmetric matrices*, Int. J. Numer. Meth. Engng., (2003), pp. 1643–1660.

[9] G. A. Geist, A. Lu, and E. L. Wachpress, *Stabilized Gaussian Reduction of an Arbitrary Matrix to Tridiagonal Form*, Tech. report, Report ORNL/TM-11089, Oak Ridge National Laboratory, TN, 1989.

[10] W. J. Givens, *Numerical Computation of the Characteristic Values of a Real Symmetric Matrix*, Tech. Report ORNL-1574, Oak Ridge National Laboratory, Oak Ridge, TN, 1954.

[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.

[12] N. J. Higham, *The Matrix Computation Toolbox*, http://www.ma.man.ac.uk/~higham/mctoolbox.

[13] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, 2002.

[14] N. J. Higham, *J-orthogonal matrices: Properties and generation*, SIAM Rev., 45 (2003), pp. 504–519.

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

[16] Z. S. Liu, *On the Extended HR Algorithm*, Technical Report PAM-564, Center for Pure and Applied Mathematics, University of California, Berkeley, CA, 1992.

[17] R. Onn, A. O. Steinhardt, and A. W. Bojanczyk, *The hyperbolic singular value decomposition and applications*, IEEE Trans. Signal Processing, 39 (1991), pp. 1575–1588.

[18] B. N. Parlett, *Reduction to tridiagonal form and minimal realizations*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 567–593.

[19] B. N. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, 1998. Corrected reprint of the 1980 original.

[20] C. M. Rader and A. O. Steinhardt, *Hyperbolic Householder transformations*, IEEE Trans. Acoust. Speech Signal Processing, ASSP-34 (1986), pp. 1589–1602.

[21] C. M. Rader and A. O. Steinhardt, *Hyperbolic Householder transforms*, SIAM J. Matrix Anal. Appl., 9 (1988), pp. 269–290.

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

[23] M. Stewart and G. W. Stewart, *On hyperbolic triangularization: Stability and pivoting*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 847–860.

[24] F. Uhlig, *The $\mathcal{DQR}$ algorithm, basic theory, convergence, and conditional stability*, Numer. Math., 76 (1997), pp. 515–553.

[25] D. Watkins and L. Elsner, *Theory of decomposition and bulge-chasing algorithms for the generalized eigenvalue problem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 943–967.

[26] R. Zurmühl and S. Falk, *Matrizen und ihre Anwendungen für angewandte Mathematiker, Physiker und Ingenieure. Teil* 2, 5th ed. Springer-Verlag, Berlin, 1986.