SIAM J. MATRIX ANAL. APPL.
Vol. 25, No. 2, pp. 301–351

AN ORTHOGONAL HIGH RELATIVE ACCURACY ALGORITHM FOR THE SYMMETRIC EIGENPROBLEM*

FROILÁN M. DOPICO†, JUAN M. MOLERA†, AND JULIO MORO†

Abstract. We propose a new algorithm for the symmetric eigenproblem that computes eigenvalues and eigenvectors with high relative accuracy for the largest class of symmetric, definite and indefinite, matrices known so far. The algorithm is divided into two stages: the first one computes a singular value decomposition (SVD) with high relative accuracy, and the second one obtains eigenvalues and eigenvectors from singular values and vectors. The SVD, used as a first stage, is responsible both for the wide applicability of the algorithm and for being able to use only orthogonal transformations, unlike previous algorithms in the literature. Theory, a complete error analysis, and numerical experiments are presented.

Key words. symmetric eigenproblem, singular value decomposition, high relative accuracy

AMS subject classifications. 65F15, 65G50, 15A18

DOI. 10.1137/10.1137/S089547980139371X

1. Introduction.

An *orthogonal spectral decomposition* of a real symmetric $n \times n$ matrix A is a factorization $A = Q\,\Lambda\,Q^T$, where Q is real orthogonal and $\Lambda = \mathrm{diag}[\lambda_1, \ldots, \lambda_n]$ is diagonal. We assume that $\lambda_1 \geq \cdots \geq \lambda_n$. The columns q_i, $i = 1, \ldots, n$, of Q are the eigenvectors of A corresponding to the eigenvalues λ_i, $i = 1, \ldots, n$. In this paper we present an algorithm that computes an orthogonal spectral decomposition for the largest class (so far) of symmetric matrices with the following *high relative accuracy*:

- The error in each computed eigenvalue, $\widehat{\lambda}_i$, is

$$|\lambda_i - \widehat{\lambda}_i| = O(\kappa\,\epsilon)|\lambda_i|, \tag{1}$$

  where we assume that $\widehat{\lambda}_1 \geq \cdots \geq \widehat{\lambda}_n$, ϵ is the unit roundoff of the finite arithmetic employed in the computation and κ is a relevant condition number, usually of order $O(1)$, to be specified later in section 2.1.
- The angle $\Theta(q_i, \widehat{q}_i)$ between each computed eigenvector $\widehat{q}_i$ and the exact one q_i satisfies

$$\Theta(q_i, \widehat{q}_i) = \frac{O(\kappa\,\epsilon)}{relgap^*(|\lambda_i|)}, \tag{2}$$

  where

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

  and the index set $\mathcal{S}$ is equal to $\{1, \ldots, n\}$ unless the eigenvalue, say λ_{j_0}, whose

*Received by the editors August 10, 2001; accepted for publication (in revised form) by I.C.F. Ipsen March 3, 2003; published electronically August 19, 2003. This research was partially supported by the Ministerio de Ciencia y Tecnología of Spain through grant BFM-2000-0008.
http://www.siam.org/journals/simax/25-2/39371.html
†Departamento de Matemáticas, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Spain (dopico@math.uc3m.es, molera@math.uc3m.es, jmoro@math.uc3m.es).

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 with the sign of λ_{j_0} satisfying $|\lambda_{j_0} - \lambda_k| \leq O(\kappa\epsilon)|\lambda_{j_0}|$. In plain words, we remove from $\mathcal{S}$ the indices corresponding to eigenvalues with opposite sign to λ_i whose absolute value is closest to $|\lambda_i|$.

Expression (2) depends on the quantity $relgap^*$, not on the eigenvalue relative gap

$$relgap(\lambda_i) = \min\left\{\min_{j\neq i} \frac{|\lambda_j - \lambda_i|}{|\lambda_i|},\, 1\right\} \tag{3}$$

one would naturally expect. The reason is that the eigenvectors are computed via the singular value decomposition (SVD), which is closely related to the spectral decomposition for symmetric matrices. Postprocessing the singular vectors produces eigenvectors with the accuracy (2). At the cost of worsening this bound in a few cases, the error in the eigenvectors can be written in terms of (3): we will show in section 5 that the error is

$$\Theta(q_i, \widehat{q}_i) = \frac{O(\kappa\,\epsilon)}{relgap(\lambda_i)} \tag{4}$$

except in the case when λ_i and λ_{j_0}, the eigenvalue whose absolute value is closest to $|\lambda_i|$, have opposite sign, and $|\lambda_{j_0}|$ is much closer to $|\lambda_i|$ than any other $|\lambda_j|$ with $\lambda_j\lambda_i > 0$. In that case,

$$\Theta(q_i, \widehat{q}_i) = \frac{O(\kappa\,\epsilon)}{\min\{relgap(\lambda_{j_0}), relgap(\lambda_i)\}}. \tag{5}$$

For the sake of simplicity, both bounds (4) and (5) have been presented in their simplest forms, when no clusters of eigenvalues with close absolute values are present. General bounds, valid in the presence of clusters, will be derived in section 5 for bases of invariant subspaces.

Equations (1), (2) may allow a considerably more accurate outcome than that of a conventional eigenvalue method, such as QR, divide-and-conquer, or bisection with inverse iteration. Such algorithms produce results with high *absolute* accuracy, i.e., satisfying

$$|\lambda_i - \widehat{\lambda}_i| = O(\epsilon)\, \max_j |\lambda_j|,$$

instead of (1), and

$$\Theta(q_i, \widehat{q}_i) = \frac{O(\epsilon)}{\frac{\min_{j\neq i} |\lambda_i - \lambda_j|}{\max_j |\lambda_j|}},$$

instead of (2). Thus, a conventional algorithm may provide approximations for the small eigenvalues ($\frac{\max_j |\lambda_j|}{|\lambda_i|} \sim \frac{1}{\epsilon}$) with no correct significant digits, or even with the wrong sign. Moreover, if there are two or more small eigenvalues, their eigenvectors may be computed very inaccurately, even when the eigenvalues are well separated in the relative sense (e.g., $\lambda_i = 10^{-15}$ and $\lambda_j = 10^{-16}$ if $\lambda_1 = 1$). At present, high relative accuracy can be reached only for certain classes of *symmetric* matrices.

Identifying classes of matrices for which either an SVD or a spectral decomposition can be computed with high relative accuracy has been a very active area of

research in the last 15 years (see [6] and references therein for an overview). So far, high relative accuracy eigensolvers are available only for some symmetric matrices and are far less developed than accurate SVD algorithms (except, of course, in the related positive definite case [7]). To be more precise, several easily characterized classes of matrices have been identified in [6] for which high relative accuracy SVDs can be computed, while present symmetric indefinite eigensolvers deliver high relative accuracy for matrices which are not easy to recognize (with the exception of scaled diagonally dominant matrices [2]). As can be seen in [22, 27], the symmetric indefinite matrices deserving high relative accuracy spectral decompositions have been characterized through the structure of their positive semidefinite polar factors. This structure, however, is very difficult to relate with the structure of the matrix itself. In this regard, *the main contribution of the present paper is to prove that the proposed eigensolver achieves high relative accuracy* (1), (2) *for all symmetric matrices in any of the classes identified in* [6]. Moreover, it will do so, under very general assumptions, for any class of matrices eventually identified in the future for which high relative accuracy SVDs can be computed. To our knowledge, none of the present symmetric eigensolvers can guarantee high relative accuracy for the classes of matrices above.

The basic motivation for the algorithm we propose is to take advantage of the present knowledge of several classes of matrices for which an SVD can be computed with high relative accuracy (see [6] for a unified approach). The connection with our work lies in that the SVD and the spectral decomposition are closely related for symmetric[1] matrices: the singular values are the absolute values of the eigenvalues, and eigenvectors may be constructed from singular vectors. To be more precise, let $A = U\Sigma V^T$ be an SVD of $A = A^T$, where U, V are $n \times n$ orthogonal with columns u_i, v_i, $i = 1, \ldots, n$, and $\Sigma = \mathrm{diag}[\sigma_1, \ldots, \sigma_n]$ with $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. In the simplest (and most frequent) case in which all singular values of A are distinct, the eigenvalues of A are

$$(v_i^T u_i)\, \sigma_i, \tag{6}$$

with $v_i^T u_i = \pm 1$ for all $i = 1, \ldots, n$, and the corresponding eigenvectors are v_i (u_i may be equally chosen). Hence, once an SVD is known, the only additional work to obtain the eigenvalues is to determine the sign ± 1 via the scalar product $v_i^T u_i$ of right and left singular vectors (the general case when groups of equal singular values appear is discussed in section 3.1). Notice that $v_i^T A v_i = v_i^T u_i \sigma_i$; i.e., the scalar product above can be thought of as a cheaper and indirect way of obtaining the sign from the Rayleigh quotient, avoiding the multiplication by the matrix A, which may give the wrong sign due to its large condition number (one example of this difficulty will be shown at the end of section 3.3). In fact, this particular way of assigning the signs through $v_i^T u_i$, together with the proof of its accuracy, is one of the crucial issues in this paper.

Therefore, given a computed high relative accuracy SVD of $A = A^T$ satisfying

$$|\sigma_i - \widehat{\sigma}_i| = O(\kappa\epsilon)\, |\sigma_i|, \tag{7}$$

$$\Theta(v_i, \widehat{v}_i) = \frac{O(\kappa\epsilon)}{relgap(\sigma_i)}, \qquad \Theta(u_i, \widehat{u}_i) = \frac{O(\kappa\epsilon)}{relgap(\sigma_i)}, \tag{8}$$

[1] All the results in this paper are valid for Hermitian matrices, although for the sake of simplicity we restrict the discussion to the real symmetric case.

with

$$relgap(\sigma_i) = \min\left\{\min_{j\neq i}\frac{|\sigma_i-\sigma_j|}{\sigma_i},\, 1\right\}, \tag{9}$$

if we prove that the pair $\widehat{v}_i$, $\widehat{u}_i$ approximates the pair v_i, u_i closely enough so that the computed value of the scalar product approximates ± 1 with an *absolute* error smaller than one (notice that this is no longer a high relative accuracy problem), then we will achieve the accuracy (1). For the eigenvectors this naive approach leads to $\Theta(q_i, \widehat{q}_i) = O(\kappa\epsilon)/relgap(\sigma_i)$, which can be improved to (2) by processing the singular vectors as described in section 5.

In this spirit we propose the following two-stage procedure to compute the eigenvalues and eigenvectors of a symmetric matrix:

Stage 1. Compute an SVD of A with accuracy (7) and (8).

Stage 2. Compute the eigenvalues of A by giving signs, according to (6), to the singular values computed in Stage 1. The corresponding eigenvectors are the right (or left) singular vectors computed in Stage 1. When groups of equal singular values are present, this step becomes more involved (see section 3.3 below).

We will show that Stage 2 provides high relative accuracy in the eigenvalues (1) and in the eigenvectors (2) as long as Stage 1 gives an SVD with small backward multiplicative error (as in formula (17) below, that in turn guarantees (7) and (8)). As to Stage 1, there are at present algorithms to perform it for several classes of matrices, summarized in [6]. These are the algorithms we are going to use, although any future high relative accuracy SVD algorithm may be employed for Stage 1.

One of the most remarkable contributions of Demmel et al. in [6] is the development of algorithms which compute high relative accuracy SVDs (i.e., satisfying (7) and (8)) for *any* matrix such that a so-called *rank-revealing decomposition* (RRD) can be computed with enough accuracy. An RRD of $G \in \mathbb{R}^{m\times n}$, $m \geq n$, is a factorization $G = \mathcal{X}\mathcal{D}\mathcal{Y}^T$ with $\mathcal{D} \in \mathbb{R}^{r\times r}$ diagonal and nonsingular and $\mathcal{X} \in \mathbb{R}^{m\times r}$, $\mathcal{Y} \in \mathbb{R}^{n\times r}$, where both matrices $\mathcal{X}$, $\mathcal{Y}$ have full column rank and are well conditioned (notice that this implies $r = \mathrm{rank}(G)$). According to the structure of the algorithms in [6], a more specific description of the *signed SVD (SSVD) algorithm* we propose here is the following.

ALGORITHM 1. (SSVD)

`Input:` Symmetric matrix A.

`Output:` Eigenvalues $\Lambda = \mathrm{diag}[\lambda_i]$ and eigenvectors $Q = [q_1 \ldots q_n]$; $A = Q\Lambda Q^T$.

1. Compute an RRD factorization XDY^T of A.
2. Compute SVD $XDY^T = U\Sigma V^T$ of RRD using algorithms from [6, section 3].
3. Compute the eigenvalues and eigenvectors of A from singular values and singular vectors using Algorithm 3 (see section 5).

We warn the reader that, before presenting Algorithm 3, we will discuss a simpler implementation of step 3 of Algorithm 1 which follows straightforwardly the ideas explained after (6). This procedure, Algorithm 2 (see section 3.3), is introduced for the sake of clarity; understanding Algorithm 3 is not easy starting from scratch, but it is elementary once the error analysis for Algorithm 2 is done in section 4. We will see there that Algorithm 2 delivers the announced accuracy (1) for eigenvalues but, in some cases, computes eigenvectors less accurately than (2). However, the error bound we obtain for eigenvectors suggests a modification in the eigenvector computation which, maintaining the validity of the error analysis, improves the accuracy in the eigenvectors to (2). This modification leads to Algorithm 3. We stress that both

versions compute the same eigenvalues and differ only in the eigenvector computation step, which is more accurate for Algorithm 3.

The accuracy required in [6] on the *computed* RRD matrices X, D, Y to guarantee that a high relative accuracy SVD can be obtained is given by the following forward error bounds:

$$\begin{aligned} |d_{ii} - \delta_{ii}| &= O(\epsilon)|\delta_{ii}|, \\ \|X - \mathcal{X}\| &= O(\epsilon)\|\mathcal{X}\|, \\ \|Y - \mathcal{Y}\| &= O(\epsilon)\|\mathcal{Y}\|, \end{aligned} \tag{10}$$

where $\|\cdot\|$ denotes the spectral norm and d_{ii}, δ_{ii} denote, respectively, the diagonal elements of D, $\mathcal{D}$. Once an RRD factorization XDY^T satisfying (10) is available, either Algorithm 3.1 or Algorithm 3.2 of [6] provides a high relative accuracy SVD of XDY^T with overall relative error (including the initial factorization stage) of order $O\left(\epsilon \max\{\kappa(X), \kappa(Y)\}\right)$ in the singular values, and $O\left(\epsilon \max\{\kappa(X), \kappa(Y)\}\right)$ over the relative gap (9) in the singular vectors, where $\kappa(\cdot)$ denotes the condition number in the spectral norm. The key to proving this high relative accuracy is that both the error (10) in the factorization and the errors introduced either by Algorithm 3.1 or by Algorithm 3.2 of [6] produce a backward error of multiplicative type, instead of the additive type usually produced by conventional algorithms (see section 2.1 for a more detailed discussion).

Several classes of matrices have been found in the last 10 years for which it is possible to compute an accurate RRD. They include bidiagonal, acyclic, Cauchy, Vandermonde, graded, and scaled diagonally dominant matrices, as well as all well-scalable symmetric positive definite matrices, some well-scalable symmetric indefinite matrices, and many others. Hence, for all symmetric matrices in any of the classes described in [6, pp. 26–27], Algorithm 1 will produce a spectral decomposition with the high relative accuracy given by (1) and (2) under the criteria given in [6] for computing accurate RRDs.

So far, the only general algorithm to compute high relative accuracy spectral decompositions of symmetric indefinite matrices is the so-called *implicit J-orthogonal* algorithm. It was introduced by Veselić in [26] and carefully analyzed by Slapničar in [22]. This algorithm begins by computing a *symmetric indefinite factorization* SJS^T of the matrix $A = A^T$, where J is square diagonal with diagonal elements ± 1, and S has full column rank.[2] If this factorization is computed with enough accuracy, the J-orthogonal algorithm yields the eigenvalues with relative error of order $O(\tilde{\kappa}\epsilon)$ for an appropriate condition number $\tilde{\kappa}$ which has been observed in practice to be of order $O(1)$. The eigenvectors are computed with error

$$\Theta(q_i, \widehat{q}_i) = \frac{O(\tilde{\kappa}\epsilon)}{relgap(\lambda_i)} \tag{11}$$

depending on the natural eigenvalue relative gap (3). This accuracy is better than the one obtained by Algorithm 1 in those cases in which the eigenvalue sign distribution is the one described right before (5). This is an advantage with respect to the algorithm proposed here. However, it should be stressed that, in view of both (4) and (5), whenever Algorithm 1 computes an eigenvector with error bound larger than the bound for

[2]Notice that, although SJS^T is not an RRD, its computation is equivalent to computing a symmetric RRD of the form XDX^T; see [23].

the J-orthogonal one, it must be due to the presence of some small eigenvalue relative gap. Thus, some other eigenvector is computed by the J-orthogonal algorithm with an error bound of similar magnitude. An illustrative example displaying this behavior will be shown in Experiment 4 in section 6.2.

An important advantage of Algorithm 1 over the J-orthogonal algorithm is that the latter does not guarantee high relative accuracy for the classes of symmetric matrices discussed in [6]. The reason is that RRDs with the accuracy (10) are obtained in [6] via Gaussian elimination with complete pivoting (GECP).[3] Moreover, a plain implementation of GECP does not guarantee accuracy (10) for most of the classes in [6]. This can be achieved only through special, nontrivial implementations of GECP, sometimes demanding a great deal of ingenuity (see [6, 5]). Since GECP leads, in general, to RRDs with $X \neq Y$, even if the matrix to be factorized is symmetric, the J-orthogonal algorithm cannot be directly applied because it begins with the *symmetric* indefinite factorization. Numerical experiments show that the usual algorithm [23] to compute the symmetric indefinite factorization does not provide, in general, the required accuracy for the symmetric matrices in those classes demanding special implementations of GECP. At present it is not known whether some modifications in the algorithm for the symmetric indefinite factorization would ensure that it is accurately computed in the sense of (10) for these matrices.

There are other important differences between the algorithm by Veselić and Slapničar and the one proposed below: the J-orthogonal algorithm uses *hyperbolic* transformations [17, section 12.5.4], which complicates the error analysis and increases the constants in the error bounds. The algorithm we propose here uses only *orthogonal* transformations. Also, the error bounds for the hyperbolic J-orthogonal algorithm are valid modulo a minor proviso (bounded growth of the scaled condition number of certain matrices appearing in each step of the iteration), while the new algorithm can be implemented in such a way that no proviso is needed to guarantee the error bounds. On the other hand, the J-orthogonal algorithm may be easily extended to matrix pencils, while this is not possible for the one presented here. There are also similarities: both algorithms require a previous factorization of the matrix, and both crucially depend on employing algorithms of one-sided Jacobi type.

Notice that the nonsymmetric character of Algorithm 1 is responsible both for making it valid for a large class of matrices and for being able to use only orthogonal transformations in step 2. The price to pay, however, is that by applying an SVD algorithm (valid for any matrix) to a symmetric matrix, we are not making any use of the symmetry of A. Thus, the algorithm is not backward stable, in the sense that one cannot guarantee that the computed eigenvalues and eigenvectors are the exact eigenvalues and eigenvectors of a close *symmetric* matrix. This is why Algorithm 1 produces an error bound in the eigenvectors which does not depend on the relative gap between the eigenvalues. This does not happen if we use a symmetric algorithm (such as the J-orthogonal algorithm) producing a *symmetric* backward error, since in that case the relative perturbation theory for symmetric matrices [16, 20, 27] leads to (11).

Concerning the computational cost of Algorithm 1, it is $O(n^3)$ provided the initial RRD costs $O(n^3)$ (some classes of matrices allow an accurate RRD, but not at $O(n^3)$ cost [6]). As is usual for high accuracy algorithms, Algorithm 1 is more expensive

[3] Some mention is also made in [6] of using QR with complete pivoting. This would open the possibility of using Algorithm 3.3 of [6], which is less costly than Algorithms 3.1–3.2 for step 2 of Algorithm 1.

than other $O(n^3)$ conventional eigenvalue methods, such as QR, divide-and-conquer, etc. The most expensive part of Algorithm 1 is the one-sided Jacobi method employed in step 2. However, some ways have been recently found [14] to speed up one-sided Jacobi which make it nearly as fast as the QR algorithm for SVD.

It is difficult to compare the cost of Algorithm 1 with that of the J-orthogonal algorithm. If in both cases we do not count the initial factorization, the difference between Algorithm 3.1 of [6] and Algorithm 3.3.1 of [22] seems to amount to two matrix multiplications and one QR factorization. However, numerical experience indicates that Algorithm 3.1 of [6] requires less Jacobi sweeps than Algorithm 3.3.1 of [22] (see section 6.2). Finally, step 3 of Algorithm 1 costs, in general, $O(n^2)$, but for every cluster with d close singular values corresponding to eigenvalues of both signs, and if eigenvectors need to be computed, there is an overhead cost of $O(d^3)+O(d^2n)$. Clearly, this is maximized when only one cluster of size $d = n$ is present. Then, the cost of step 3 is $O(n^3)$. As to the timing statistics, the run-times of both algorithms are comparable according to the numerical experiments below.

Both the comments on the computational cost and the numerical experiments in section 6.2 apply to a plain implementation of the one-sided Jacobi SVD algorithm included in Algorithm 3.1 of [6]. At present, fast and sophisticated implementations of the one-sided Jacobi SVD algorithm are being developed by Z. Drmač along the lines of [14]. We have tested a preliminary version of this routine in a few numerical experiments, and with this optimized Jacobi, Algorithm 1 was much faster than the J-orthogonal algorithm. Extensive numerical experiments will be done in the future.

The rest of the paper is organized as follows. Section 2 collects the mathematical results required to perform a complete error analysis of Algorithm 1. Section 3 describes in detail Algorithm 2, a preliminary implementation for step 3 of Algorithm 1, including the corresponding pseudocode. Section 4 contains a complete error analysis of a first, simpler implementation of Algorithm 1, using Algorithm 2 in step 3. This is done in the most general setting, allowing for the presence of clusters, which is why an entire section is devoted to discussing the error analysis. Otherwise, if the matrix has well-separated singular values, the error analysis is straightforward. We remind the reader that there are two reasons for doing the error analysis on this preliminary implementation: first, this error analysis gives the idea of how to design the final Algorithm 3 for step 3 of Algorithm 1. The second reason is that, once the error analysis is done with Algorithm 2, no new error analysis is required for Algorithm 3. Section 5 is devoted to developing and analyzing Algorithm 3, proving the error bounds (2), (4), and (5) in the most general setting, with any distribution of clusters. To keep the presentation within limits, most of the proofs in section 5 have been omitted (see [10] for complete proofs). However, in order to give a hint of the ideas and techniques employed we include in an appendix the proof of Theorem 5.7, one of the main results in section 5. Section 6 addresses the practical implementation of Algorithm 1, together with the numerical tests. Conclusions and discussion of open problems are presented in section 7.

2. Preliminary results. We collect in this section the mathematical results required to perform the error analysis of Algorithm 1. As stated in the introduction, the only requirement on the high relative accuracy SVD algorithm in step 2 of Algorithm 1 is producing a small multiplicative backward error when performed in finite arithmetic. A precise statement is given in section 2.1 for algorithms in [6]. We also show in section 2.1 that the error due to the initial RRD can be absorbed as an additional multiplicative backward error. Section 2.2 summarizes the multiplicative

perturbation theory for singular values and for bases of singular subspaces needed to guarantee the high relative accuracy of the overall algorithm.

2.1. Backward error of the SVD algorithm. The following theorem is essentially proved in [6].

THEOREM 2.1. *Algorithm* 3.1 *of* [6] *(see Algorithm* 4 *in section* 6.1 *below) produces a multiplicative backward error when executed with machine precision* ϵ*; i.e., if* $G = XDY^T \in \mathbb{R}^{m\times n}$ *is the RRD computed in step* 1 *of Algorithm* 1 *and* $\widehat{U}\widehat{\Sigma}\widehat{V}^T$ *is the SVD computed by the algorithm, then there exist matrices* $U' \in \mathbb{R}^{m\times r}$, $V' \in \mathbb{R}^{n\times r}$, $E \in \mathbb{R}^{m\times m}$, $F \in \mathbb{R}^{n\times n}$ *such that* U' *and* V' *have orthonormal columns,*

$$\begin{aligned} \|U' - \widehat{U}\| = O(\epsilon), \quad \|V' - \widehat{V}\| = O(\epsilon), \\ \|E\| = O(\epsilon\kappa(X)), \quad \|F\| = O(\epsilon\kappa(R')\kappa(Y)), \end{aligned} \tag{12}$$

where R' *is the best conditioned row diagonal scaling of the triangular matrix* R *appearing in* step 1 *of Algorithm* 3.1 *of* [6] *and*

$$(I + E)G(I + F) = U'\widehat{\Sigma}V'^T. \tag{13}$$

It is proved in [6] that $\kappa(R')$ is at most of order $O(n^{3/2}\kappa(X))$, but in practice we have observed in extensive numerical tests that $\kappa(R')$ behaves as $O(n)$. One can get rid of the factor $\kappa(R')$ at the price of using the more costly Algorithm 3.2 of [6].

We state Theorem 2.1 because the original result [6, Thm. 3.1] is not phrased as a backward error result, which is what we need for the subsequent error analysis. The only missing piece in the analysis of [6] is the fact that one-sided Jacobi [17, section 8.6.3] produces a small multiplicative backward error. This can be easily derived from Proposition 3.13 in [13] and, since it is not central to our argument, we omit its proof, together with that of Theorem 2.1. A full proof of both results will appear elsewhere [11] (and can be found in [10, Appendix A]). Two different versions of Algorithm 3.1 of [6] are analyzed in [11], depending on whether the right- or left-handed version of one-sided Jacobi is employed. One can show that the right-handed version, i.e., the one in which the Jacobi rotations are applied from the right, guarantees smaller error bounds and leads precisely to Theorem 2.1. For the left-handed version one can prove a result similar to Theorem 2.1, but with a weaker error bound for F, and requiring a minor proviso to ensure the accuracy. However, the left-handed version is still the one usually employed in practice since it is much faster and no significant difference has ever been observed in accuracy. This is why we use it in most of the experiments in section 6. Finally, it is crucial for the accuracy of one-sided Jacobi algorithms to impose as a stopping criterion that the cosines of the angles between the different columns (or rows, depending on the version of one-sided Jacobi) be smaller than ϵ times the dimension of the matrix.

Once the backward error of the SVD algorithm is shown to be multiplicative, the perturbation theory in section 2.2 below can be used to prove high relative accuracy, namely that the computed singular values and vectors of XDY^T satisfy

$$\begin{aligned} |\sigma_i - \widehat{\sigma}_i| &= O(\kappa\,\epsilon)\,\sigma_i, \\ \Theta(v_i, \widehat{v}_i) &= \frac{O(\kappa\,\epsilon)}{relgap(\sigma_i)}, \\ \Theta(u_i, \widehat{u}_i) &= \frac{O(\kappa\,\epsilon)}{relgap(\sigma_i)}, \end{aligned} \tag{14}$$

where

$$\kappa = \kappa(R') \max\{\kappa(X), \kappa(Y)\} \tag{15}$$

is the relevant condition number announced in the introduction.

As a matter of fact, one may even absorb into a backward error of the form (13) the error in the initial RRD, i.e., the one due to the fact that the SVD computation does not start from the symmetric matrix A itself but from its computed RRD: let $A = \mathcal{X}\mathcal{D}\mathcal{Y}^T$ be an exact RRD factorization of A and assume the starting decomposition XDY^T has been computed accurately enough so that the computed matrices X, D, Y satisfy conditions (10). Then, as shown in the proof of Theorem 2.1 in [6], there exist matrices E_f, F_f with $\|E_f\| = O(\epsilon\kappa(X))$, $\|F_f\| = O(\epsilon\kappa(Y))$ such that

$$(I + E_f)A(I + F_f) = XDY^T. \tag{16}$$

This, together with (13), implies that

$$U'\widehat{\Sigma}V'^T = (I + \widetilde{E})A(I + \widetilde{F}), \tag{17}$$

where the backward errors $\widetilde{E}$, $\widetilde{F}$ are of size $\|\widetilde{E}\| = O(\epsilon\kappa(X))$, $\|\widetilde{F}\| = O(\epsilon\kappa(R')\kappa(Y))$ and reflect that the errors produced by both the RRD factorization and the SVD algorithm are backward multiplicative.

We stress that all our error analysis is done in terms of the backward errors $\|\widetilde{E}\|$ and $\|\widetilde{F}\|$. Although we have focused on the case when $\|E_f\| = O(\epsilon\kappa(X))$ and $\|F_f\| = O(\epsilon\kappa(Y))$, any other more general backward errors for the factorization step can be trivially incorporated into the error analysis, since, up to first order,

$$\|\widetilde{E}\| \le \|E_f\| + O(\epsilon\kappa(X)), \quad \|\widetilde{F}\| \le \|F_f\| + O(\epsilon\kappa(R')\kappa(Y)).$$

2.2. Multiplicative perturbation theory. Let G be a real $m \times n$ matrix with SVD $G = U\Sigma V^T$ and singular values $\sigma_1 \ge \sigma_2 \ge \cdots$. We consider a multiplicative perturbation $\widetilde{G} = (I + E)G(I + F)$ of G with SVD $\widetilde{G} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^T$ and singular values $\widetilde{\sigma}_i$, also ordered decreasingly.

THEOREM 2.2 (exactly Theorem 3.1 of [16]). *Let* $G \in \mathbb{R}^{m\times n}$, $\widetilde{G} = (I+E)G(I+F)$, *and set*

$$\eta = \max\{\|E\|, \|F\|\}, \qquad \eta' = 2\eta + \eta^2. \tag{18}$$

Then

$$\frac{|\,\sigma_i - \widetilde{\sigma}_i|}{\sigma_i} \le \eta'.$$

In addition to the change in the singular values, we also need to estimate the changes undergone by singular subspaces or, more precisely, by their bases. Although the following results are valid for rectangular matrices (see [20, 8]), we state them in the square case, the only case we deal with in section 4. Thus, G is now a real $n \times n$ matrix and $\widetilde{G} = (I + E)G(I + F)$. Let

$$G = [U_1\ U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \tag{19}$$

$$\widetilde{G} = \begin{bmatrix} \widetilde{U}_1\ \widetilde{U}_2 \end{bmatrix} \begin{bmatrix} \widetilde{\Sigma}_1 & 0 \\ 0 & \widetilde{\Sigma}_2 \end{bmatrix} \begin{bmatrix} \widetilde{V}_1^T \\ \widetilde{V}_2^T \end{bmatrix} \tag{20}$$

be two conformally partitioned SVDs of G and $\widetilde{G}$; i.e., the four matrices $\Sigma_1, \widetilde{\Sigma}_1 \in \mathbb{R}^{q\times q}$ and $\Sigma_2, \widetilde{\Sigma}_2 \in \mathbb{R}^{(n-q)\times(n-q)}$ are diagonal. No particular order is assumed on the singular values. The change in the singular subspaces is usually measured through the sines of the canonical angles $\Theta(U_1, \widetilde{U}_1)$ between the column spaces of U_1 and $\widetilde{U}_1$, and $\Theta(V_1, \widetilde{V}_1)$ between the column spaces of V_1 and $\widetilde{V}_1$ (see [25]). It is well known that this change is governed (see, e.g., [20, Thm. 4.1]) by the singular value relative gap

$$rg(\Sigma_1, \widetilde{\Sigma}_2) = \min_{\substack{\sigma\in\sigma(\Sigma_1)\\ \widetilde{\sigma}\in\sigma(\widetilde{\Sigma}_2)}} \frac{|\sigma - \widetilde{\sigma}|}{\widetilde{\sigma}}, \tag{21}$$

where $\sigma(M)$ denotes the set of singular values of the matrix M.

This kind of result, however, is not enough for our purposes. The fact that the signs of the eigenvalues are obtained through scalar products like the one in (6) forces us to accurately compute not only the singular subspaces but also the corresponding *simultaneous* bases U_i and V_i. To ensure this, finer perturbation results are needed, dealing with the sensitivity of the bases themselves. It has been observed in [8] that simultaneous bases of singular subspaces do not have the same sensitivity under perturbation as their corresponding singular subspaces. More precisely, bases may be much more sensitive to *additive* perturbations than singular subspaces. Fortunately enough for our purposes, both sensitivities are essentially equal for multiplicative perturbations. A detailed discussion of these issues may be found in [8, 9], including a stronger version of the following result (we use the Frobenius norm $\|\cdot\|_F$, as is usual when the dimension of the subspaces is larger than 1).

THEOREM 2.3 (exactly Theorem 2.2 of [8]). *Let $G \in \mathbb{R}^{n\times n}$ and $\widetilde{G} = (I + E)G(I + F)$ with respective SVDs* (19) *and* (20). *Then there exists an orthogonal matrix $P \in \mathbb{R}^{q\times q}$ such that*

$$\sqrt{\|U_1P - \widetilde{U}_1\|_F^2 + \|V_1P - \widetilde{V}_1\|_F^2} \le 2\sqrt{q}\left[\eta + \frac{\eta'}{1-\eta}\ \frac{1}{relgap(\Sigma_1, \widetilde{\Sigma}_2)}\right], \tag{22}$$

where $relgap(\Sigma_1, \widetilde{\Sigma}_2)$ is given by

$$relgap(\Sigma_1, \widetilde{\Sigma}_2) = \min\{rg(\Sigma_1, \widetilde{\Sigma}_2), 1\}, \tag{23}$$

and η, η' are given by (18).

Although it is more usual in the literature [6, 5] to define the relative gap (21) with the roles of Σ_1 and Σ_2 reversed, we have chosen the definition above to conform to the cited perturbation theorems. However, this does not represent any significant difference in the error bounds, since a straightforward calculation shows that

$$2\, relgap(\widetilde{\Sigma}_2, \Sigma_1) \ge relgap(\Sigma_1, \widetilde{\Sigma}_2) \ge \frac{1}{2}\, relgap(\widetilde{\Sigma}_2, \Sigma_1). \tag{24}$$

On the other hand, as is usual in this kind of perturbation bounds, one can reformulate the definition of the gaps to make them depend only on the unperturbed singular values, at the cost of somewhat complicating the bounds.

The main point of Theorem 2.3 is that the orthogonal matrix P is the same for both left and right singular vectors. This will be enough to guarantee the accuracy of the sign assignment and of the computed bases of invariant subspaces.[4]

3. Computing spectral decompositions from SVDs. This section is divided into three parts. Section 3.1 outlines the mathematical basis for the main idea underlying Algorithm 1, namely that one can easily get a spectral decomposition of a symmetric matrix if one is given an SVD, even if the matrix has groups of equal singular values. Some practical details concerning clusters of close singular values in finite precision will be considered in section 3.2. The complete pseudocode for Algorithm 2 will be presented in section 3.3. This is the simplest implementation of step 3 in Algorithm 1.

3.1. Mathematical basis. Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with SVD $A = U\Sigma V^T$. Then, $V^TAV = V^TU\Sigma$ is orthogonally similar to A with V^TU orthogonal. If A has distinct singular values $\sigma_1 > \sigma_2 > \cdots > \sigma_p$ with respective multiplicities m_i, $i = 1, \ldots, p$ ($m_1 + \cdots + m_p = n$), and we partition U and V accordingly as

$$U = \left[\ \mathcal{U}_1 \mid \mathcal{U}_2 \mid \cdots \mid \mathcal{U}_p\ \right],$$
$$V = \left[\ \mathcal{V}_1 \mid \mathcal{V}_2 \mid \cdots \mid \mathcal{V}_p\ \right]$$

with $\mathcal{U}_i, \mathcal{V}_i \in \mathbb{R}^{n\times m_i}$ corresponding to each distinct singular value, then

$$\mathcal{V}_i^T\mathcal{U}_j = 0 \quad \text{whenever} \quad i \neq j \tag{25}$$

since, due to the symmetry of A, both its left and right singular vectors are eigenvectors of A^2. Consequently,

$$V^TU = \operatorname{diag}[\mathcal{V}_1^T\mathcal{U}_1, \ldots, \mathcal{V}_p^T\mathcal{U}_p] \tag{26}$$

is block-diagonal, where each diagonal block $\mathcal{V}_i^T\mathcal{U}_i \in \mathbb{R}^{m_i\times m_i}$ is itself orthogonal. Furthermore, since

$$V^TAV = \operatorname{diag}[\sigma_1\mathcal{V}_1^T\mathcal{U}_1, \ldots, \sigma_p\mathcal{V}_p^T\mathcal{U}_p] \tag{27}$$

is symmetric, we conclude that each $\mathcal{V}_i^T\mathcal{U}_i$ is not only orthogonal but also symmetric and its eigenvalues, ± 1, are precisely the signs of those eigenvalues of A having modulus σ_i. In the simplest case when $m_i = 1$, the eigenvalue is just $v_i^Tu_i\,\sigma_i$. In the general case, a simple calculation shows that if the spectrum of $\mathcal{V}_i^T\mathcal{U}_i$ contains m_i^+ eigenvalues equal to 1 and m_i^- equal to -1 ($m_i = m_i^+ + m_i^-$), then

$$m_i^{\pm} = \frac{m_i \pm \operatorname{trace}(\mathcal{V}_i^T\mathcal{U}_i)}{2}; \tag{28}$$

i.e., the multiplicity of the eigenvalues $\pm\sigma_i$ can be easily recovered from the trace of $\mathcal{V}_i^T\mathcal{U}_i$.

[4] Actually, Theorem 2.3 is stronger than the usual bounds on the canonical angles between singular subspaces, since one can easily show that $\|\sin(\Theta(U_1, \widetilde{U}_1))\|_F \leq \|U_1\,P - \widetilde{U}_1\|_F$, which holds similarly for V_1.

To obtain the eigenvectors of A, the simplest (and more frequent) case corresponds to $m_i = 1$. In that case, the right singular vector v_i itself is an eigenvector. When some m_i is larger than 1 and $\mathrm{trace}(\mathcal{V}_i^T\mathcal{U}_i) = m_i$ (resp., $\mathrm{trace}(\mathcal{V}_i^T\mathcal{U}_i) = -m_i$), the m_i eigenvalues are all equal to σ_i (resp., $-\sigma_i$), and the eigenvectors are the columns of $\mathcal{V}_i$. In the general case $m_i > 1$, $m_i \neq m_i^{\pm}$, consider for each $i = 1, \ldots, p$ an orthogonal diagonalization of $\mathcal{V}_i^T\mathcal{U}_i = \mathcal{W}_i J_i \mathcal{W}_i^T$, with $J_i = \mathrm{diag}[I_{m_i^+}, -I_{m_i^-}]$ and $\mathcal{W}_i = [\mathcal{W}_i^+ \,|\, \mathcal{W}_i^-] \in \mathbb{R}^{m_i \times m_i}$ partitioned conformally to J_i. Then, denoting $\mathcal{W} = \mathrm{diag}[\mathcal{W}_1, \ldots, \mathcal{W}_p]$, one can easily check that the matrix $Q = V\mathcal{W}$ is such that

$$Q^T A Q = \mathrm{diag}[\sigma_1 J_1, \ldots, \sigma_p J_p];$$

i.e., the set of columns of the submatrix $Q_i^+ = \mathcal{V}_i\mathcal{W}_i^+ \in \mathbb{R}^{n \times m_i^+}$ (resp., $Q_i^- = \mathcal{V}_i\mathcal{W}_i^- \in \mathbb{R}^{n \times m_i^-}$) is a basis of the eigenspace corresponding to the eigenvalue σ_i (resp., $-\sigma_i$) of A. In other words, $A = Q\Lambda Q^T$ with $\Lambda = \mathrm{diag}[\pm\sigma_i]$ is a spectral decomposition of A.

We conclude by noting that, although the right singular vectors $\mathcal{V}_i$ have been used throughout the argument, the symmetry of A implies that similar results hold using instead the left singular vectors $\mathcal{U}_i$.

3.2. Clusters in finite arithmetic. We have seen how to deal theoretically with groups of equal singular values. When working in finite precision, however, it is unlikely that some of the singular values in the output of step 2 of Algorithm 1 come out equal. But at the same time the expected accuracy (14) determines that some of the singular values should be considered as numerically indistinguishable and treated in the spirit of section 3.1. Thus we are forced to deal with, say, k different groups Σ_i of n_i close singular values ($i = 1, \ldots, k$, $n_1 + \cdots + n_k = n$), which we call *clusters*.[5] The criterion to divide the singular values into clusters is crucial for the final accuracy of Algorithm 1. This criterion will be carefully analyzed in section 4.4, where we show that to achieve the accuracy (1) (see Theorem 4.3) it is enough to include two contiguous singular values σ_j, σ_{j+1} in the same cluster whenever

$$\frac{|\sigma_j - \sigma_{j+1}|}{\sigma_j} \leq C\,\kappa\,\epsilon \tag{29}$$

for a suitable constant C, where

$$\kappa = \kappa(R')\max\{\kappa(X), \kappa(Y)\}$$

is the quantity (15) which came up in the error bound for the singular values computed in step 2 of Algorithm 1 (see section 4.4 for more on the choice of the constant C; we mention here that in the performed numerical experiments the choice $C = 1$ always gives very satisfactory results).

For each cluster Σ_i we take matrices U_i, $V_i \in \mathbb{R}^{n \times n_i}$ whose columns are, respectively, left and right singular vectors corresponding to the singular values in Σ_i. Since the singular values in Σ_i are, in general, different, each U_i and V_i is made up with several of the matrices $\mathcal{U}_j$ and $\mathcal{V}_j$ defined in section 3.1. Consequently, the products $\Delta_i = V_i^T U_i$ are symmetric, orthogonal, and block-diagonal matrices whose diagonal blocks are some of the blocks $\mathcal{V}_j^T\mathcal{U}_j$.

[5] For the sake of brevity, we use Σ_i to denote both the cluster of singular values and the corresponding $n_i \times n_i$ diagonal matrix.

We conclude by noting that the numbers n_i^+ of positive and n_i^- of negative eigenvalues with absolute values in the cluster Σ_i are still given by a formula such as (28). As to the eigenvectors, things are different from section 3.1, since the diagonalization of Δ_i does not lead, in general, to eigenvectors but just to two orthonormal bases, one for the invariant subspace corresponding to the positive eigenvalues in the cluster Σ_i and another for the negative ones. This is a fundamental issue in the error analysis for the eigenvector computations and will be carefully explained throughout the proof of Theorem 4.4.

3.3. A first version of step 3 of Algorithm 1. In this section we describe Algorithm 2, the first implementation of step 3 in Algorithm 1. The eigenvalue and the eigenvector computations are separated in the procedure into two independent parts. Doing this helps us to better understand the structure of Algorithm 3, our final implementation of step 3 in Algorithm 1, which will only insert a different cluster selection routine in between the eigenvalue and the eigenvector computations.

ALGORITHM 2.

Input: SVD of a symmetric matrix $A = U\Sigma V^T$.
Output: Eigenvalues $\Lambda = \mathrm{diag}[\lambda_i]$ and eigenvectors $Q = [q_1 \dots q_n]$; $A = Q\Lambda Q^T$.

1. Decide the singular value clusters, $\Sigma_i = \{\sigma_{i_0}, \dots, \sigma_{i_0+n_i-1}\}$, U_i, V_i, $i = 1, \dots, k$, according to (29).
2. Compute the eigenvalues using Algorithm 2.1 below.
3. Compute the eigenvectors using Algorithm 2.2 below.

ALGORITHM 2.1.

Input: SVD of $A = U\Sigma V^T$; Clusters $\Sigma_1, \Sigma_2, \dots, \Sigma_k$.
Output: Eigenvalues Λ.

1. **for** each cluster, $i = 1 : k$
2. compute the diagonal elements of $\Delta_i = V_i^T U_i$
3. **if** $n_i = 1$ **then**
4. $\lambda_{i_0} = sign(\Delta_i)\sigma_{i_0}$
5. **else**
6. **for** $j = i_0 : i_0 + n_i - 1$
7. $\lambda_j = sign[(\Delta_i)_{jj}]\sigma_j$
8. **endfor**
9. $t_i = \mathrm{trace}(\Delta_i)$, $n_i^- = \frac{n_i - t_i}{2}$
10. **if** $\#\{(\Delta_i)_{jj} < 0\} \neq n_i^-$ **then**
11. **for** $j = i_0 : i_0 + n_i^- - 1$
12. $\lambda_j = -\sigma_j$
13. **endfor**
14. **for** $j = i_0 + n_i^- : i_0 + n_i - 1$
15. $\lambda_j = \sigma_j$
16. **endfor**
17. **endif**
18. **endif**
19. **endfor**

ALGORITHM 2.2.

Input: SVD of $A = U\Sigma V^T$; Clusters $\Sigma_1, \Sigma_2, \dots, \Sigma_k$; Eigenvalues Λ.
Output: Eigenvectors $Q = [q_1 \dots q_n]$.
Notation: $Q_i^{\pm}$ denotes the eigenvector matrix corresponding to positive (resp., negative) eigenvalues in Σ_i.

```
1. for each cluster, i = 1 : k
2.     if n_i = 1 then
3.         q_{i_0} = v_{i_0}
4.     else
5.         n_i^- ≡ number of negative eigenvalues in Σ_i
6.         if n_i^- = 0 then
7.             Q_i^+ = V_i
8.         elseif n_i^- = n_i then
9.             Q_i^- = V_i
10.        else
11.            multiply Δ_i = V_i^T U_i
12.            diagonalize Δ_i = [W_i^+ W_i^-] J_i [W_i^+ W_i^-]^T
13.            Q_i^+ = V_i W_i^+,   Q_i^- = V_i W_i^-
14.        endif
15.    endif
16. endfor
```

Some comments on this code are in order. First, we have singled out the case $n_i = 1$, although it is not needed. This is done to highlight the fact that Algorithm 2 is extremely simple in this case, with all complications coming from the case $n_i > 1$.

Notice also that the code does not compute eigenvectors associated with zero eigenvalues in the case where $r = \mathrm{rank}(A) < n$. This is due to the fact that the SVD algorithms in [6] do not compute null vectors. However, if accurate null vectors are needed, they can be obtained as the last $n - r$ columns of the orthogonal factor in a complete QR factorization of the matrix V of right singular vectors.

If large clusters are present, one can save flops in steps 11 and 13 of Algorithm 2.2 by employing Strassen multiplication without spoiling the accuracy of the overall algorithm. As to the diagonalization step, step 12 of Algorithm 2.2, it is assumed that one performs it on a symmetrization of Δ_i. This is crucial to obtain orthonormal eigenvectors.

Notice that the eigenvalue sign assignment (steps 6–17 of Algorithm 2.1) is done in two stages when there are clusters: First (steps 6–8), we assign the signs given by the diagonal elements of $\Delta_i = V_i^T U_i$ as if the singular values in Σ_i were not a cluster. If the number of assigned negative eigenvalues coincides with $n_i^- = \frac{n_i - \mathrm{trace}(\Delta_i)}{2}$, the signs are kept. Otherwise, we proceed as described in steps 10–17 of Algorithm 2.1. The reason for this is that the random sign assignment inside each cluster in steps 10–17 proved to be too pessimistic in practice: although singular values inside each cluster are numerically indistinguishable according to (14), actual errors are frequently smaller than the error bounds. These smaller errors are lost if the signs of eigenvalues are randomly assigned. The modified procedure minimizes this loss of accuracy.

We finish this section with an interesting remark on the way the signs are assigned in Algorithm 2. One might think of obtaining the sign of each eigenvalue from the Rayleigh quotients $v_i^T A v_i$, one of the most common ways of approximating eigenvalues, instead of from $v_i^T u_i$. However, it is easy to construct examples for which the sign of $v_i^T A v_i$ is wrong, while the sign of $v_i^T u_i$ is right. We propose the following numerical example, easily reproducible in MATLAB 5.3: Generate a 100×100 symmetric Cauchy matrix with parameters $x_i = y_i \equiv r_i$, $i = 1 : 100$, where r_i is a random number chosen from a normal distribution with mean zero and variance one. Scale this matrix on both sides by the same diagonal matrix with diagonal elements $d_i = 10^{20 r_i'}$, where r_i' is a random number chosen from a uniform distribution on the interval $(0.0, 1.0)$. For

matrices of this kind Algorithm 3 in [5] can be used to obtain in a very simple way an RRD, $A = XDY^T$, with forward errors fulfilling (10). Finally, applying Algorithm 3.1 of [6] to this RRD yields an SVD of A with high relative accuracy. No clusters of singular values are present in general. For several of the computed singular vectors neither $v_i^T A v_i$ nor $(v_i^T X)D(Y^T v_i)$ have the same sign of $v_i^T u_i$, which is the correct one, as will be shown in section 4 (the reader also can check this by using a symbolic package such as Mathematica in very high precision). This example shows that using Rayleigh quotients may be dangerous, even in the case when the matrix is given as an RRD. Similar behavior is not rare in other Cauchy matrices or in random RRDs with very ill-conditioned diagonals. The use of Rayleigh quotients in the more favorable case when the matrix A is scaled in a certain particular way is covered in [15].

4. Error analysis. In this section we present the rounding error analysis for the eigenvalues and the eigenvectors computed by Algorithm 1 using Algorithm 2 in step 3. This error analysis remains valid for Algorithm 1 using Algorithm 3 in step 3: this is trivially true for the eigenvalues, since both versions of Algorithm 1 compute the same eigenvalues. It is also true for the eigenvectors, due to the generality of the error analysis, which allows us to use the new clusters of singular values appearing in Algorithm 3.

We stress that the error analysis applies *to the entire* Algorithm 1, since it relies on the backward multiplicative error formula (17), which absorbs the errors of the initial factorization in step 1. Although we focus on the case when the RRD is computed with the error (10), which ensures $\|E_f\| = O(\epsilon\kappa(X))$ and $\|F_f\| = O(\epsilon\kappa(Y))$, any other more general backward errors for the factorization step can be trivially incorporated into the error analysis, as explained at the end of section 2.1.

The main results in this section are the forward error bounds in Theorems 4.3 and 4.7. Both are expressed in big-O notation, without explicitly specifying the dimensional constants involved. There are two reasons for this. First, we rely on error bounds in [6], which are written in big-O notation without explicit mention of the constants. Second, it is well known that the precise value of the constant is, in general, not relevant for practical purposes.

This said, the reader should be aware that in the statements of the theorems in this section we absorb moderately growing functions of the dimensions (either n, of the whole matrix, or n_i, of the clusters) as constants inside the $O(\kappa\epsilon)$. Since none of them exceeds a moderate number times n^2, we choose not to write them explicitly in order not to complicate further the error bounds. However, the interested reader may find those corresponding to step 3 of Algorithm 1 explicitly stated in the proofs.

The error analysis is performed in the most general case when clusters of singular values are present. This somewhat complicates the analysis, which is almost straightforward in the simple (and most likely) case of matrices whose singular values are distinct enough. The practical criterion to decide when two singular values belong to the same cluster is also discussed in detail.

In the rest of this section we only deal with the error in nonzero eigenvalues and the corresponding eigenvectors. If the original matrix is singular, the number of zero eigenvalues is determined exactly, provided an RRD factorization fulfilling (10) is computed. As to the null vectors, it can be shown that they can be computed with error $O(\epsilon\,\kappa(R')\,\max\{\kappa(X), \kappa(Y)\})$ using the method already described following Algorithm 2.2. The relative gap does not appear because in this case it is equal to one.

We begin by fixing our model for floating point arithmetic and the notation.

4.1. Model of arithmetic. We use the conventional error model for floating point arithmetic,

$$\texttt{fl}(a \odot b) = (a \odot b)(1+\delta), \tag{30}$$

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 occurs. We stress that the results proved in this section still hold under a weaker error model valid for arithmetic with no guard digit.

The error analysis below also remains valid for complex Hermitian matrices, since [18, Chapter 3] the equality (30) continues to hold for complex numbers with δ a small complex number bounded by $|\delta| = O(\epsilon)$. However, in order to simplify the presentation we consider only real symmetric matrices.

Finally, we will commit a slight abuse of notation, denoting by $\texttt{fl}(expr)$ the computed result in finite precision of expression $expr$, instead of its rigorous meaning of the closest floating point number to $expr$.

4.2. Notation. Letters with hats denote computed quantities appearing in any step of Algorithm 1. The same letters without hats denote their exact counterparts. It is assumed that the input of Algorithm 1 is a real symmetric $n \times n$ matrix A, for which an RRD factorization XDY^T with small multiplicative backward error (16) can be computed.

We assume that k different clusters $\widehat{\Sigma}_i$ of n_i $(n_1 + \cdots + n_k = n)$ close singular values are identified through criterion (29); thus, the usual decreasing order on singular values determines the unknown exact clusters Σ_i. The singular values of one particular cluster are supposed to be different from the singular values of any other cluster. Given an index $i \in \{1, \ldots, k\}$, we define

$$\Sigma_{\bar{i}} = \bigcup_{j \neq i} \Sigma_j. \tag{31}$$

For each cluster Σ_i we take matrices U_i, $V_i \in \mathbb{R}^{n \times n_i}$ whose columns are, respectively, left and right singular vectors corresponding to the singular values in Σ_i. Recall that the singular values in Σ_i may be different, so both U_i and V_i will, in general, contain singular vectors corresponding to different singular values. Therefore, the remarks in section 3.2 apply.

Many nontrivial choices are possible for the exact quantities U_i, V_i if A has multiple singular values in Σ_i. In that case, the results proved in this section are valid for *any* possible choice of U_i and V_i, provided their columns are singular vectors and not simply bases of the corresponding singular subspaces.

4.3. Fundamental lemma. The following lemma, which is a simple consequence of the fundamental perturbation theorem, Theorem 2.3, and the multiplicative backward error formula (17) for steps 1 and 2 of Algorithm 1, is the starting point of our error analysis. For the sake of brevity, the quantities K_i will be defined inside Lemma 4.1. These quantities play a relevant role in the error analysis.

LEMMA 4.1. *Let $\widehat{U}_i, \widehat{V}_i \in \mathbb{R}^{n \times n_i}$ be the matrices of computed left and right singular vectors corresponding to the cluster of singular values $\widehat{\Sigma}_i$ computed by steps* 1–2 *of Algorithm* 1 *applied to the symmetric matrix A. Let U_i, V_i, Σ_i be their exact*

counterparts. Then, there exists an exact orthogonal matrix P_i such that

$$K_i \equiv \sqrt{\|U_iP_i - \widehat{U}_i\|_F^2 + \|V_iP_i - \widehat{V}_i\|_F^2} \leq \frac{O(\kappa\,\epsilon)}{relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})} \tag{32}$$

with κ given by (15)*.*

Proof. Let U_i', V_i' be the submatrices corresponding to $\widehat{\Sigma}_i$ of the exact orthogonal matrices U' and V' appearing in (17). Then, Theorem 2.3 applied to (17) guarantees that there exists an orthogonal $n_i \times n_i$ matrix P_i such that

$$\sqrt{\|U_iP_i - U_i'\|_F^2 + \|V_iP_i - V_i'\|_F^2} = \left\| \begin{bmatrix} U_iP_i - U_i' \\ V_iP_i - V_i' \end{bmatrix} \right\|_F \leq \frac{O(\kappa\,\epsilon)}{relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})}.$$

Notice that

$$\sqrt{\|U_iP_i - \widehat{U}_i\|_F^2 + \|V_iP_i - \widehat{V}_i\|_F^2} = \left\| \begin{bmatrix} U_iP_i - \widehat{U}_i \\ V_iP_i - \widehat{V}_i \end{bmatrix} \right\|_F,$$

so the triangular inequality implies

$$\sqrt{\|U_iP_i - \widehat{U}_i\|_F^2 + \|V_iP_i - \widehat{V}_i\|_F^2} \leq \left\| \begin{bmatrix} U_iP_i - U_i' \\ V_iP_i - V_i' \end{bmatrix} \right\|_F + \left\| \begin{bmatrix} U_i' - \widehat{U}_i \\ V_i' - \widehat{V}_i \end{bmatrix} \right\|_F.$$

The last term in the right-hand side of this inequality is $O(\epsilon)$ by (12). This concludes the proof. ∎

Lemma 4.1 gives a forward error bound for simultaneous orthonormal bases of singular subspaces, which depends only on the quantities $\|\widetilde{E}\|$ and $\|\widetilde{F}\|$ appearing in (17). In other words, it only accounts for errors corresponding to steps 1 and 2 of Algorithm 1, i.e., to the SVD computation.

The rest of the bounds obtained in this section, i.e., those corresponding to step 3 of Algorithm 1, depend, for each cluster, on the quantities K_i on the left-hand side of (32). This allows us to write all subsequent error bounds as a function of K_i and to trace how each of the steps in Algorithm 2 contributes to the final error. From now on we assume that all quantities K_i for $i = 1, \ldots, k$ are sufficiently smaller than 1, which, according to Lemma 4.1, is the case whenever the clusters of singular values are properly chosen. More precisely, all we need is that K_i be small enough to make all bounds in sections 4.4 and 4.5 strictly smaller than one.

4.4. Error bounds for eigenvalues and cluster criterion. We begin by analyzing the error produced in the computation of $\mathrm{trace}(V_i^T U_i)$ using the standard inner product algorithm.

LEMMA 4.2. *Let $\widehat{U}_i, \widehat{V}_i \in \mathbb{R}^{n \times n_i}$ be the matrices of computed left and right singular vectors corresponding to the cluster of singular values $\widehat{\Sigma}_i$ computed by steps* 1–2 *of Algorithm* 1 *applied to the symmetric matrix A. Let U_i, V_i, Σ_i be their exact counterparts. Then,*

$$\begin{aligned} \left| \mathtt{fl}(\,\mathrm{trace}(\mathtt{fl}(\,\widehat{V}_i^T \widehat{U}_i\,))\,) - \mathrm{trace}(\,V_i^T U_i\,) \right| &\leq \sqrt{n_i}\left(\sqrt{2}K_i + \frac{K_i^2}{2}\right) + O(\epsilon) \\ &\leq \frac{O(\kappa\,\epsilon)}{relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})} \end{aligned} \tag{33}$$

with κ given by (15) *and K_i by* (32)*.*

Proof. First observe that

$$\left|\mathtt{fl}(\,\mathrm{trace}(\mathtt{fl}(\,\widehat{V}_i^T\widehat{U}_i\,))\,) - \mathrm{trace}(\,V_i^TU_i\,)\right| \le \left|\mathtt{fl}(\,\mathrm{trace}(\mathtt{fl}(\,\widehat{V}_i^T\widehat{U}_i\,))\,) - \mathrm{trace}(\,\widehat{V}_i^T\widehat{U}_i\,)\right| + \left|\mathrm{trace}(\,\widehat{V}_i^T\widehat{U}_i\,) - \mathrm{trace}(\,V_i^TU_i\,)\right|. \tag{34}$$

Taking into account that the norm of the columns of $\widehat{U}_i$ and $\widehat{V}_i$ is close to one by (12), a straightforward error analysis [18, Chapter 3] shows that the first term in the right-hand side of inequality (34) is $n_i(n+n_i)\epsilon + O(\epsilon^2)$. If P_i is the orthogonal matrix appearing in Lemma 4.1, the last term fulfills

$$\begin{aligned}\left|\mathrm{trace}(\,\widehat{V}_i^T\widehat{U}_i\,) - \mathrm{trace}(\,V_i^TU_i\,)\right| &= \left|\mathrm{trace}(\,\widehat{V}_i^T\widehat{U}_i\,) - \mathrm{trace}(\,P_i^TV_i^TU_iP_i\,)\right| \\ &\le \sqrt{n_i}\sqrt{\sum_{k=1}^{n_i}\left|(\,\widehat{V}_i^T\widehat{U}_i - P_i^TV_i^TU_iP_i\,)_{kk}\right|^2} \\ &\le \sqrt{n_i}\,\|\widehat{V}_i^T\widehat{U}_i - (V_iP_i)^TU_iP_i\|_F. \end{aligned} \tag{35}$$

Now define matrices Δ_u and Δ_v such that

$$\widehat{U}_i = U_iP_i + \Delta_u \quad \text{and} \quad \widehat{V}_i = V_iP_i + \Delta_v. \tag{36}$$

Combining (35) and (36) yields

$$\left|\mathrm{trace}(\,\widehat{V}_i^T\widehat{U}_i\,) - \mathrm{trace}(\,V_i^TU_i\,)\right| \le \sqrt{n_i}\,(\,\|\Delta_u\|_F + \|\Delta_v\|_F + \|\Delta_u\|_F\|\Delta_v\|_F\,),$$

where we have used that $\|CD\|_F \le \|C\|_2\|D\|_F$ for any matrices C, D, together with the fact that the spectral norm of any matrix with orthonormal columns is one. Finally, setting $K_i = \sqrt{\|\Delta_u\|_F^2 + \|\Delta_v\|_F^2}$ as in (32), we obtain, after some direct manipulations, the desired result. ∎

Notice that $\mathrm{trace}\left(V_i^TU_i\right)$ may only take the integer values $-n_i, -n_i+2, \ldots, n_i-4, n_i-2, n_i$, since $V_i^TU_i$ is symmetric and orthogonal. Thus, it is sufficient that the error bound in (33) be less than one to compute *exactly* the value of $\mathrm{trace}\left(V_i^TU_i\right)$. This can be done by obtaining t_i, the nearest integer to $\mathtt{fl}(\,\mathrm{trace}(\,\mathtt{fl}(\,\widehat{V}_i^T\widehat{U}_i\,)\,)\,)$ *with the parity of* n_i. Then, the *integer* computation (with integer variables) of $(n_i - t_i)/2$ yields n_i^-, the *exact number of negative eigenvalues* included in the cluster Σ_i of singular values. The exact number of positive eigenvalues is obtained from the integer computation of $n_i - n_i^-$.

We stress that the conditions

$$\left|\mathtt{fl}(\,\mathrm{trace}(\mathtt{fl}(\,\widehat{V}_i^T\widehat{U}_i\,)\,)\,) - \mathrm{trace}(\,V_i^TU_i\,)\right| < 1, \qquad i = 1, \ldots, k, \tag{37}$$

which ensure that signs are correctly assigned, determine the cluster criterion to be used in Algorithm 2. Giving a rigorous criterion would require an exact knowledge of the constants involved in the big-O bound in (33), which in any case are too pessimistic in practice. Instead, we assume that the singular values in each cluster $\widehat{\Sigma}_i$ satisfy

$$relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}}) \approx relgap(\widehat{\Sigma}_i, \widehat{\Sigma}_{\bar{i}}) > C\epsilon\kappa(R')\max(\kappa(X), \kappa(Y))$$

for a suitable constant C. This can be obtained by defining that two contiguous singular values $\widehat{\sigma}_j \ge \widehat{\sigma}_{j+1}$ belong to the same cluster whenever

$$\frac{|\widehat{\sigma}_j - \widehat{\sigma}_{j+1}|}{\widehat{\sigma}_j} \le C\,\kappa\,\epsilon,$$

i.e., whenever condition (29) above holds. Choosing a large C ensures (37) and, as a consequence, that the number of positive/negative eigenvalues is correctly computed. However, a large value for C favors the mixing of different singular values in the same cluster and, since the signs are assigned more or less randomly within each cluster, the error bound in the eigenvalues becomes roughly the product of C times the bound in the singular values (see (14)). Therefore, the choice of C is subject to a certain trade-off. A sensible choice might be choosing C between 1 and 10. All the numerical experiments in section 6 have been done with $C = 1$ and the results are very satisfactory.

In any case, notice that, on one hand, the singular values are computed with the accuracy given by (17) and Theorem 2.2. On the other hand, their signs as eigenvalues of A are correctly assigned whenever the bound (33) is less than one. With this we have proved the main result of this subsection.

THEOREM 4.3. *Let A be an $n \times n$ real symmetric matrix for which it is possible to compute an RRD fulfilling* (10). *Let $\lambda_1 \geq \cdots \geq \lambda_n$ be the eigenvalues of A and $\widehat{\lambda}_1 \geq \cdots \geq \widehat{\lambda}_n$ be the approximations to the eigenvalues of A computed by Algorithm* 1. *Let $\widehat{U}_i, \widehat{V}_i \in \mathbb{R}^{n \times n_i}$ be the matrices of computed left and right singular vectors corresponding to the cluster of computed singular values $\widehat{\Sigma}_i$, and let U_i, V_i, Σ_i be their exact counterparts. Assume that all clusters have been chosen according to* (29), *so that conditions* (37) *hold. Then*

$$|\lambda_j - \widehat{\lambda}_j| = |\lambda_j| \; O(\epsilon \kappa(R') \max(\kappa(X), \kappa(Y))), \qquad j = 1, \ldots, n. \tag{38}$$

The error bound (38) holds even for zero eigenvalues, since the *exact* number of zero eigenvalues of A is known once an RRD factorization satisfying (10) is available.

4.5. Error bounds for eigenvectors. In this section we obtain bounds on the distance between *bases* of invariant subspaces. Although it is more common to bound the sines of the canonical angles between the exact and the computed invariant subspaces [25], we choose to compare the bases themselves because, as explained before Theorem 2.3, bases play an essential role both in Algorithm 2 and in its error analysis. However, usual $\sin \Theta$ bounds easily follow from Theorem 4.7, since distances between bases and canonical angles between subspaces are closely related [25, Thms. I.5.2 and II.4.11] and the same bounds hold for both, up to a factor $\sqrt{2}$ in Frobenius norm.

One important issue in the subsequent analysis comes from step 12 of Algorithm 2.2 in which the $n_i \times n_i$ matrix $\widehat{V}_i^T \widehat{U}_i$ is orthogonally diagonalized for each cluster $\widehat{\Sigma}_i$. Lemma 4.1 shows that the matrices $\widehat{U}_i, \widehat{V}_i$ of computed singular vectors are not reliable approximations of the matrices of exact singular vectors U_i, V_i, but just reliable approximations of $U_i P_i$ and $V_i P_i$, with P_i the unknown $n_i \times n_i$ orthogonal matrix in Lemma 4.1. Hence, we are forced in practice to diagonalize approximations to matrices $P_i^T V_i^T U_i P_i$. Theorem 4.4 shows that this is enough to get orthonormal bases of invariant subspaces, although not for obtaining eigenvectors.

THEOREM 4.4. *Let A be a symmetric $n \times n$ matrix and $U_i, V_i \in \mathbb{R}^{n \times n_i}$ be matrices of left and right singular vectors of A corresponding to a cluster of nonzero singular values Σ_i, different from the rest of the singular values of A. Let P_i be any $n_i \times n_i$ orthogonal matrix, and consider any orthogonal diagonalization of the $n_i \times n_i$ orthogonal and symmetric matrix $P_i^T V_i^T U_i P_i$ partitioned as*

$$P_i^T V_i^T U_i P_i = [W_i^+ \; W_i^-] \begin{bmatrix} I_{n_i^+} & 0 \\ 0 & -I_{n_i^-} \end{bmatrix} [W_i^+ \; W_i^-]^T, \tag{39}$$

where I_s denotes the $s \times s$ identity matrix and $n_i^+ + n_i^- = n_i$. Then the columns of $V_i P_i W_i^+$ (resp., $V_i P_i W_i^-$) form an orthonormal basis of the invariant subspace of A corresponding to the positive (resp., negative) eigenvalues whose absolute values are in Σ_i.

Proof. Without loss of generality, we may consider the SVD of A partitioned in only two blocks,

$$A = [U_1\ U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} [V_1\ V_2]^T, \tag{40}$$

where no special order is assumed on the singular values. Here Σ_1 corresponds to the cluster Σ_i to be studied and Σ_2 corresponds to the remaining clusters $\Sigma_{\bar{i}}$ defined as in (31). The matrix P_i will be denoted just by P, and the matrices $W_i^{\pm}$ in (39) will be denoted by $W_{\pm}$.

As mentioned in section 3.2, $V_1^T U_1$ is orthogonal, symmetric, and block-diagonal with the size of the blocks fixed by the groups of equal singular values inside Σ_1. The matrix $V_1^T U_1 \Sigma_1$ is also symmetric with the same block-diagonal structure of $V_1^T U_1$. An orthogonal diagonalization for each block of $V_1^T U_1$ leads to an orthogonal diagonalization of the full matrix $V_1^T U_1$ with eigenvectors which are also eigenvectors of $V_1^T U_1 \Sigma_1$. In this situation, the eigenvectors of $V_1^T U_1$ corresponding to the eigenvalue 1 (resp., -1) are the eigenvectors of $V_1^T U_1 \Sigma_1$ corresponding to positive (resp., negative) eigenvalues with absolute values in Σ_1. From this we deduce that the invariant subspaces corresponding to positive (resp., negative) eigenvalues of matrices $P^T V_1^T U_1 P$ and $P^T V_1^T U_1 \Sigma_1 P$ coincide. Once this is taken into account, the rest of the proof reduces to some easy block manipulations.

Combining (40) and $V_2^T U_1 = 0$ from (25), we obtain

$$AV_1P = U_1\Sigma_1 P = [V_1\ V_2] \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} U_1\Sigma_1 P = V_1 P (P^T V_1^T U_1 \Sigma_1 P). \tag{41}$$

Splitting the spectrum into positive and negative eigenvalues, we orthogonally diagonalize

$$P^T V_1^T U_1 \Sigma_1 P = [Q_+\ Q_-] \begin{bmatrix} D_+ & 0 \\ 0 & D_- \end{bmatrix} [Q_+\ Q_-]^T,$$

and from (41) we obtain

$$A(\,V_1PQ_+\,) = (\,V_1PQ_+\,)D_+ \quad \text{and} \quad A(\,V_1PQ_-\,) = (\,V_1PQ_-\,)D_-. \tag{42}$$

Now, we know that $\mathrm{col}(Q_{\pm}) = \mathrm{col}(W_{\pm})$, and since the columns of $Q_{\pm}$ and $W_{\pm}$ are orthonormal, there exist square orthogonal matrices $T_{\pm}$ such that $W_{\pm} = Q_{\pm} T_{\pm}$. Combining this and (42) we obtain

$$A(\,V_1PW_{\pm}\,) = (\,V_1PW_{\pm}\,)\,(\,T_{\pm}^T D_{\pm} T_{\pm}\,),$$

which proves the theorem. ∎

Once the previous theorem is proved, the rest of the section is organized into the following three steps.

1. Although Lemma 4.1 guarantees that $\widehat{U}_i$ and $\widehat{V}_i$ are close to U_iP_i and V_iP_i, provided the clusters have been properly chosen, this does not mean that $\widehat{\Delta}_i =$

$\mathtt{fl}(\widehat{V}_i^T\widehat{U}_i)$ in step 11 of Algorithm 2.2 is symmetric. Let $\widehat{S}_i$ be the symmetric matrix obtained by replacing the upper triangular part of $\widehat{\Delta}_i$ with its lower triangular part. Lemma 4.5 bounds the difference between $\widehat{S}_i$ and the exact symmetric matrix $P_i^T V_i^T U_i P_i$. Notice that if any driver routine of LAPACK [1] for the symmetric eigenvalue problem is used in step 12 of Algorithm 2.2, just the upper (or lower) triangular part of $\widehat{\Delta}_i$ is stored. Hence, the symmetrization step does not require any additional work.

2. Lemma 4.6 relates the computed orthogonal eigendecomposition of $\widehat{S}_i$ with an exact eigendecomposition of $P_i^T V_i^T U_i P_i$. It is shown that exact matrices $W_i^{\pm}$ in (39) can be chosen close enough to the corresponding computed matrices $\widehat{W}_i^{\pm}$ in step 12 of Algorithm 2.2.

3. Finally, the main theorem, Theorem 4.7, bounds the difference between the $n \times n_i^{\pm}$ matrices $\mathtt{fl}(\widehat{V}_i\widehat{W}_i^{\pm})$ computed in step 13 of Algorithm 2.2 and some orthonormal bases of exact invariant subspaces of A. This result is a simple consequence of Lemmas 4.1 and 4.6.

The bottom line after these three steps is that step 3 of Algorithm 1 produces errors of the order of K_i, the quantity defined in (32), whose upper bound (32) depends only on the errors in steps 1 and 2 of Algorithm 1.

LEMMA 4.5. *Let $\widehat{U}_i, \widehat{V}_i \in \mathbb{R}^{n\times n_i}$ be the matrices of computed left and right singular vectors corresponding to the cluster of singular values $\widehat{\Sigma}_i$ computed by steps* 1–2 *of Algorithm* 1 *applied to the symmetric matrix A. Let U_i, V_i, Σ_i be their exact counterparts. Let $\widehat{S}_i$ be a symmetrization of the floating point matrix $\widehat{\Delta}_i = \mathtt{fl}(\,\widehat{V}_i^T\widehat{U}_i\,)$ obtained by replacing the upper triangular part of $\widehat{\Delta}_i$ with its lower triangular part, or vice versa. Then an orthogonal $n_i \times n_i$ matrix P_i exists such that*

$$\begin{aligned} \|\widehat{S}_i - P_i^T V_i^T U_i P_i\|_F &\le 2K_i + \frac{K_i^2}{\sqrt{2}} + O(\epsilon) \\ &\le \frac{O(\epsilon\kappa(R')\max(\kappa(X),\kappa(Y)))}{relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})}. \end{aligned} \tag{43}$$

Proof. First observe that

$$\|\mathtt{fl}(\,\widehat{V}_i^T\widehat{U}_i\,) - P_i^T V_i^T U_i P_i\|_F \le \|\mathtt{fl}(\,\widehat{V}_i^T\widehat{U}_i\,) - \widehat{V}_i^T\widehat{U}_i\|_F + \|\widehat{V}_i^T\widehat{U}_i - P_i^T V_i^T U_i P_i\|_F,$$

where P_i is the orthogonal matrix appearing in Lemma 4.1. Standard error analysis of usual matrix multiplication [18], and the fact that the columns of $\widehat{U}_i$ and $\widehat{V}_i$ are almost orthonormal by (12), show that the first term in the right hand-side of the previous inequality is bounded by $n\, n_i\epsilon + O(\epsilon^2)$. The last term can be bounded as in the proof of Lemma 4.2, so we obtain

$$\|\mathtt{fl}(\,\widehat{V}_i^T\widehat{U}_i\,) - P_i^T V_i^T U_i P_i\|_F \le \left(\sqrt{2}K_i + \frac{K_i^2}{2}\right) + O(\epsilon).$$

We write $\mathtt{fl}(\,\widehat{V}_i^T\widehat{U}_i\,) = \widehat{L} + \widehat{D} + \widehat{R}$ as the sum of its strict lower triangular part, its diagonal part, and its strict upper triangular part. The same is done for the symmetric matrix $P_i^T V_i^T U_i P_i = L + D + L^T$, so the previous equation yields

$$\sqrt{\|(\widehat{L}+\widehat{D}) - (L+D)\|_F^2 + \|\widehat{R} - L^T\|_F^2} \le \left(\sqrt{2}K_i + \frac{K_i^2}{2}\right) + O(\epsilon). \tag{44}$$

The same inequality holds for $\sqrt{\|\widehat{L}-L\|_F^2+\|\widehat{D}+\widehat{R}-(D+L^T)\|_F^2}$. On the other hand

$$\|\widehat{S}_i - P_i^T V_i^T U_i P_i\|_F = \sqrt{\|(\widehat{L}+\widehat{D})-(L+D)\|_F^2+\|\widehat{L}^T-L^T\|_F^2}.$$

Combining this equation with (44) proves the lemma. ∎

Errors in the diagonalization step, step 12, of Algorithm 2.2 are now analyzed. Notation and definitions of the previous lemma are used.

LEMMA 4.6. *Let $\widehat{W}_i\widehat{\Lambda}_i\widehat{W}_i^T$ be the computed orthogonal spectral decomposition of the symmetric $n_i \times n_i$ matrix $\widehat{S}_i$ using any* LAPACK *subroutine for the symmetric eigenproblem* [1, *section* 2.3.4.1]. *Then, there exists a matrix E_i, an orthogonal matrix Z_i, and an orthogonal matrix P_i such that*

$$P_i^T V_i^T U_i P_i + E_i = Z_i \widehat{\Lambda}_i Z_i^T, \tag{45}$$

where

$$\|Z_i - \widehat{W}_i\|_2 \le O(\epsilon) \quad \text{and} \quad \|E_i\|_F \le 2K_i + \frac{K_i^2}{\sqrt{2}} + O(\epsilon). \tag{46}$$

Moreover, if $\widehat{W}_i^+$ (resp., $\widehat{W}_i^-$) is the submatrix of $\widehat{W}_i$ with columns corresponding to the positive (resp., negative) elements of $\widehat{\Lambda}_i$, then there exist matrices W_i^+, W_i^- fulfilling (39) *such that*

$$\begin{aligned} \|\widehat{W}_i^{\pm} - W_i^{\pm}\|_F &\le 2\sqrt{2}K_i + K_i^2 + O(\epsilon) \\ &= \frac{O(\epsilon\kappa(R')\max(\kappa(X),\kappa(Y)))}{relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})}. \end{aligned} \tag{47}$$

Proof. Using the results in [1, section 4.7.1] we see that there exist an orthogonal matrix Z_i and a matrix E_i' such that

$$\widehat{S}_i + E_i' = Z_i \widehat{\Lambda}_i Z_i^T, \tag{48}$$

where

$$\|Z_i - \widehat{W}_i\|_2 \le O(\epsilon) \quad \text{and} \quad \|E_i'\|_2 \le O(\epsilon)\|\widehat{S}_i\|_2.$$

Let P_i be the orthogonal matrix appearing in Lemmas 4.1 and 4.5. The spectral norm of the orthogonal matrix $P_i^T V_i^T U_i P_i$ is equal to one, so (43) implies $\|\widehat{S}_i\|_2 = 1+\beta$, with $|\beta| \le 2K_i + K_i^2/\sqrt{2} + O(\epsilon)$. Thus $\|E_i'\|_2 = O(\epsilon)$. Now, expressions (45) and (46) are easily proved using Lemma 4.5, noting by (48) that

$$P_i^T V_i^T U_i P_i + \widehat{S}_i - P_i^T V_i^T U_i P_i + E_i' = Z_i \widehat{\Lambda}_i Z_i^T,$$

and defining

$$E_i = \widehat{S}_i - P_i^T V_i^T U_i P_i + E_i'.$$

We finally prove (47). Let $W_i^{\pm}$ be matrices fulfilling (39) and Z_i^+ (resp., Z_i^-) be a submatrix of Z_i corresponding to the positive (resp., negative) elements of $\widehat{\Lambda}_i$. We assume that K_i is small enough to imply $\|E_i\|_2 < 1$, so the eigenvalues equal

to 1 (resp., -1) of $P_i^T V_i^T U_i P_i$ remain positive (resp., negative) in $\widehat{\Lambda}_i$. This can be seen by applying Weyl's eigenvalue perturbation theorem to (45) (see, for instance, [25, Corollary IV.4.10]). Thus, Davis and Kahan's $\sin\Theta$ theorem for variations of invariant subspaces of Hermitian matrices [4] applied to (45) leads to

$$\|\sin\Theta(W_i^+, Z_i^+)\|_F \le \frac{\|E_i\|_F}{\min\limits_{\substack{\mu<0\\ \mu\in\widehat{\Lambda}_i}} |1-\mu|} \le \|E_i\|_F, \tag{49}$$

where the matrix $\Theta(W_i^+, Z_i^+)$ is the matrix of the canonical angles between the column space of W_i^+ and the column space of Z_i^+. Theorem II.4.11 in [25], (49), and (46) show that it is possible to choose W_i^+ such that

$$\begin{aligned}\|W_i^+ - Z_i^+\|_F &= \sqrt{\|\sin\Theta(W_i^+, Z_i^+)\|_F^2 + \|I - \cos\Theta(W_i^+, Z_i^+)\|_F^2}\\ &\le \sqrt{2}\,\|\sin\Theta(W_i^+, Z_i^+)\|_F\\ &\le \sqrt{2}\,\|E_i\|_F\\ &\le 2\sqrt{2}K_i + K_i^2 + O(\epsilon). \qquad (50)\end{aligned}$$

Similar results hold for W_i^- and Z_i^-. We finish the proof by noting that

$$\|\widehat{W}_i^{\pm} - W_i^{\pm}\|_F \le \|\widehat{W}_i^{\pm} - Z_i^{\pm}\|_F + \|Z_i^{\pm} - W_i^{\pm}\|_F.$$

The first term of the right-hand side is $O(\epsilon)$ by (46), and the second one is bounded in (50). ∎

We conclude with the main result on rounding errors for eigenvectors computed in step 13 of Algorithm 2.2. Previous notation and definitions are used.

THEOREM 4.7. *Let A be an $n\times n$ real symmetric matrix of rank r for which it is possible to compute an RRD fulfilling* (10). *Let $\widehat{\Sigma}_i$ be a cluster of nonzero computed singular values of A using steps* 1–2 *of Algorithm* 1 *and Σ_i be the corresponding cluster of exact singular values. Then there exist matrices Q_i^+ and Q_i^-, whose columns form orthonormal bases of the invariant subspaces of A corresponding, respectively, to the positive and negative eigenvalues of A with absolute values in Σ_i, such that*

$$\begin{aligned}\|\mathtt{fl}(\,\widehat{V}_i\widehat{W}_i^+\,) - Q_i^+\|_F &\le (2\sqrt{2}+1)(K_i + K_i^2) + K_i^3 + O(\epsilon)\\ &= \frac{O(\epsilon\kappa(R')\max(\kappa(X),\kappa(Y)))}{relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})}, \qquad (51)\end{aligned}$$

with an equal bound for $\|\mathtt{fl}(\,\widehat{V}_i\widehat{W}_i^-\,) - Q_i^-\|_F$.

Moreover, let $\widehat{Q} = [\mathtt{fl}(\,\widehat{V}_1\widehat{W}_1^+\,)\,\mathtt{fl}(\,\widehat{V}_1\widehat{W}_1^-\,)\ldots\mathtt{fl}(\,\widehat{V}_k\widehat{W}_k^+\,)\,\mathtt{fl}(\,\widehat{V}_k\widehat{W}_k^-\,)]$ be the $n\times r$ matrix whose columns are the bases of all considered invariant subspaces of A computed using Algorithm 1. *Then there exists an $n\times r$ matrix B with exact orthonormal columns such that*

$$\|\widehat{Q} - B\|_F = O(\epsilon). \tag{52}$$

Proof. Let $\widehat{V}_i$ be the matrix of computed right singular vectors corresponding to the cluster $\widehat{\Sigma}_i$, and let V_i be its exact counterpart. Let $W_i^{\pm}$, $\widehat{W}_i^{\pm}$, and P_i be the matrices appearing in Lemmas 4.6 and 4.1. By Theorem 4.4, the columns of $Q_i^+ \equiv V_i P_i W_i^+$ and $Q_i^- \equiv V_i P_i W_i^-$ are orthonormal bases of the invariant subspaces

of A corresponding, respectively, to the positive and negative eigenvalues of A with absolute values in Σ_i.

Note also that

$$\|\mathtt{fl}(\widehat{V}_i\widehat{W}_i^{\pm}) - V_iP_iW_i^{\pm}\|_F \leq \|\mathtt{fl}(\widehat{V}_i\widehat{W}_i^{\pm}) - \widehat{V}_i\widehat{W}_i^{\pm}\|_F + \|\widehat{V}_i\widehat{W}_i^{\pm} - V_iP_iW_i^{\pm}\|_F. \tag{53}$$

The first term of the right-hand side is bounded by $n_i\sqrt{n_i\,n_i^{\pm}}\,\epsilon + O(\epsilon^2)$ using the standard error analysis of usual matrix multiplication [18] and the fact that the columns of $\widehat{V}_i$ and $\widehat{W}_i^{\pm}$ are almost orthonormal by (12) and (46). For the second term we proceed as follows: Define matrices Δ_v and $\Delta_w^{\pm}$ by

$$\widehat{V}_i = V_iP_i + \Delta_v \quad \text{and} \quad \widehat{W}_i^{\pm} = W_i^{\pm} + \Delta_w^{\pm},$$

where $\|\Delta_v\|_F \leq K_i$ by (32) and $\|\Delta_w^{\pm}\|_F \leq 2\sqrt{2}K_i + K_i^2 + O(\epsilon)$ by (47). Thus

$$\begin{aligned}\|\widehat{V}_i\widehat{W}_i^{\pm} - V_iP_iW_i^{\pm}\|_F &\leq \|\Delta_v\|_F + \|\Delta_w^{\pm}\|_F + \|\Delta_v\|_F\|\Delta_w^{\pm}\|_F \\ &\leq (2\sqrt{2}+1)(K_i + K_i^2) + K_i^3 + O(\epsilon).\end{aligned}$$

Combining this with (53) proves (51).

Finally, (52) follows from the well-known fact that finite precision matrix multiplication of matrices with columns orthonormal up to $O(\epsilon)$ yields a matrix with columns orthonormal up to $O(\epsilon)$. ∎

As announced in the introduction, the eigenvector error bounds we derive suffer from an important drawback: they depend on *relgap* (23) between singular values, which is less than or equal to the natural relative gap between eigenvalues, the one expected for the symmetric eigenproblem. This is an unavoidable consequence of the nonsymmetric character of Algorithm 1. This drawback, however, can be partially solved applying Theorem 4.7 to certain new singular value clusters chosen as described in section 5.

It is worth observing that Theorem 4.7 does not guarantee that the columns of the matrices $\mathtt{fl}(\widehat{V}_i\widehat{W}_i^{\pm})$ computed by Algorithm 1 approximate *eigenvectors* of A. This can only be ensured in three cases: when there is no cluster ($n_i = 1$), when all eigenvalues in the cluster have the same sign, and when the cluster contains eigenvalues of both signs with either $n_i^+ = 1$ or $n_i^- = 1$. In this last case, either $\mathtt{fl}(\widehat{V}_i\widehat{W}_i^{+})$ or $\mathtt{fl}(\widehat{V}_i\widehat{W}_i^{-})$ approximates an eigenvector of A. In any other situation, the columns of $\mathtt{fl}(\widehat{V}_i\widehat{W}_i^{\pm})$ do not approximate eigenvectors but just orthonormal bases of the invariant subspaces of A corresponding to either the positive or the negative eigenvalues with absolute values in the cluster. However, provided the clusters of singular values are chosen according to criterion (29), this does not represent any drawback, because the eigenvectors in the corresponding invariant subspaces are computed by any symmetric eigensolver (including the J-orthogonal algorithm [26, 22]) with large errors due to the presence of very small relative gaps between the eigenvalues inside the clusters. There is no need to say that the J-orthogonal algorithm also computes accurate bases of invariant subspaces, due to its backward stability.

We conclude with an interesting remark concerning the discussion in the previous paragraph. Consider, for simplicity, that according to criterion (29) a cluster of two singular values, one corresponding to a positive eigenvalue and the other to a negative one, has been found. Then the bound in Theorem 4.7 implies that Algorithm 1 computes *both* eigenvectors with an error governed by the relative gap between the cluster and the singular values outside the cluster. This can be much larger than

the relative gap between the singular values inside the cluster. Thus, the presence of clusters reduces the errors in the computed eigenvectors. We will take more advantage of this property in section 5.

5. Computing more accurate eigenvectors. The error in the eigenvectors computed by Algorithm 2 is governed (see Theorem 4.7) by the singular value relative gap, which is less than or equal to the natural eigenvalue relative gap. We present in this section Algorithm 3, an implementation of step 3 of Algorithm 1, which computes eigenvectors with the error (2) (see also (4) and (5)) announced in the introduction. As we will see, the underlying idea is very simple and does not require a new error analysis but simply takes advantage of the generality of the one performed in section 4. We stress that the eigenvalue computation (steps 1–2 in Algorithm 2) will stay the same; only the computation of the eigenvectors will be modified. The general case, when clusters of singular values of arbitrary dimension are present, will be considered.

First, note that Algorithm 2 computes the eigenvalues before computing the eigenvectors. The relative error in the eigenvalues is of order $O(\epsilon\kappa(R')\max(\kappa(X),\kappa(Y)))$ provided the clusters are chosen according to criterion (29). A second important remark is that the error analysis performed in section 4 for the eigenvectors is independent of the error analysis for the eigenvalues, both being valid under the hypothesis that the quantities K_i defined in (32) are sufficiently small. As Lemma 4.1 shows, this is achieved by defining clusters which yield large enough $relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})$, but whenever this condition is fulfilled different clusters, i.e., different K_i, can be chosen to compute the eigenvectors using Algorithm 2.2. Theorem 4.7 still applies and will provide a smaller error bound whenever the new clusters *for the eigenvector computation* have larger *relgaps* than the ones chosen according to (29). Consequently we present the following algorithm that is the final version of step 3 of Algorithm 1.

ALGORITHM 3.

`Input: SVD of a symmetric matrix` $A = U\Sigma V^T$.

`Output: Eigenvalues` $\Lambda = \mathrm{diag}[\lambda_i]$ `and eigenvectors` $Q = [q_1 \dots q_n]$; $A = Q\Lambda Q^T$.

1. `Decide the singular value clusters,` $\{\Sigma_i, U_i, V_i\}_{i=1}^{k}$`, according to (29).`
2. `Compute the eigenvalues using Algorithm 2.1.`
3. `Use Algorithm 3.1 in section 5.2 to merge, when necessary, some pairs of clusters to form a new set` $\{\Sigma_i, U_i, V_i\}_{i=1}^{q}$ `of clusters, according to the strategy developed in this section.`
4. `Compute the eigenvectors using Algorithm 2.2 on the new set of clusters.`

The difference with respect to Algorithm 2 is the presence of step 3, in which a new selection of clusters is made. The limit for improving the bound (51) in Theorem 4.7 by increasing $relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})$ is naturally the eigenvalue relative gap. With this in mind, the idea to be implemented is very simple: Let Σ_i be one of the singular value clusters chosen according to (29), and let Λ_i^+ (resp., Λ_i^-) be the corresponding clusters of positive (resp., negative) eigenvalues with absolute values in Σ_i. Then $relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})$ can be much worse than the minimum of the two eigenvalue relative gaps associated to Σ_i only in the case in which Σ_i is *signed* (all the eigenvalues of the same sign), and the closest (in the relative sense) cluster to Σ_i, let us say $\Sigma_{cl(i)}$, is oppositely signed. Without loss of generality, it can be assumed that $\Sigma_i = \Lambda_i^+$; therefore $\Sigma_{cl(i)} = -\Lambda_{cl(i)}^-$. If Σ_i and $\Sigma_{cl(i)}$ are joined to form a new cluster $\Lambda_i^+ \cup (-\Lambda_{cl(i)}^-)$ with a larger *relgap*, the bound (51) will improve *separately* for the bases of *exactly the same two invariant*

subspaces associated with Λ_i^+ and $\Lambda_{cl(i)}^-$, computed by Algorithm 2.2 applied to the new set of clusters. Therefore, nothing is lost by merging clusters of this kind, and the error bound (51) can improve by *joining* close adjacent clusters in such a way that $relgap$ increases.

It will be seen that in the other cases it is not necessary to join clusters, either because the singular value relative gap is already of the same order of the eigenvalue relative gap, or because joining clusters would mean increasing the number of eigenvalues of the same sign in the cluster, and consequently Algorithm 2.2 would compute bases of a larger invariant subspace, thus losing all the information about the original invariant subspaces.

The error bound for the eigenvectors computed by Algorithm 3 is given by (51) applied to the new set of clusters chosen in step 3. The formula (2) for individual eigenvectors follows easily from (51). The argument is as follows: Consider an individual eigenvalue λ_i, positive without loss of generality, belonging to a cluster Σ_k (chosen in step 3 of Algorithm 3). If λ_i is not the only positive eigenvalue in Σ_k, then (2) follows inmediately. If λ_i is the only positive eigenvalue in Σ_k and there are other negative eigenvalues in the cluster, then (2) follows because in Theorem 4.7 the bounds for the bases associated to positive and negative eigenvalues are independent of the relative gaps between the singular values inside Σ_k. The only remaining case is the one in which $\Sigma_k = \{\lambda_i\}$, i.e., the eigenvalue is by itself a cluster. If its closest cluster has not been joined to Σ_k by step 3 of Algorithm 3, it is either because it contains positive eigenvalues or because merging the two clusters would not improve the singular value relative gap. In any case, removing the closest (in absolute value) negative eigenvalues changes the singular value relative gap at most by a moderate factor. Therefore, (2) also holds in this case.

We will also relate our sharpest bound (51) with the eigenvalue relative gap. More precisely, we will show in this section that Algorithm 3 guarantees that the error in the computed basis of the invariant subspace corresponding to each cluster of eigenvalues $\widehat{\Lambda}_i$ of the symmetric matrix A is smaller than

$$\frac{O(\epsilon\kappa(R')\max(\kappa(X),\kappa(Y)))}{\min\{relgap(\widehat{\Lambda}_i), relgap(\widehat{\Lambda}_{cl(i)})\}}, \tag{54}$$

where the eigenvalue relative gap in the denominator corresponds to either the cluster $\widehat{\Lambda}_i$ under consideration or the cluster $\widehat{\Lambda}_{cl(i)}$ whose eigenvalues have different sign but are the closest (in relative sense) in absolute value. This result will be proved in Theorem 5.12 and generalizes to invariant subspaces the error bound (4), (5) appearing in the introduction for eigenvectors.

The rest of this section is organized as follows: Some relationships between eigenvalue and singular value relative gaps are proved in section 5.1. This is necessary if (54) has to be proved using Theorem 4.7, which only deals with singular value relative gaps. First we show in Theorem 5.5 that in the case of an *unsigned* cluster (a cluster containing singular values corresponding to positive and negative eigenvalues), the singular value relative gap of the cluster is not worse, up to a moderate constant, than an eigenvalue relative gap. Theorem 5.6 proves that this also happens to the relative gap of a signed cluster if the closest cluster is not signed of the opposite sign. Thus for clusters of these two kinds (54) holds, and it is not necessary to join them to any other cluster.

In the rest of section 5.1 we will study the case of a signed cluster whose closest cluster is oppositely signed. In all the theorems it will be assumed that the singular value relative gap is sufficiently smaller than the eigenvalue relative gap; otherwise it is trivial that (54) is reached. With these assumptions (54) is always achieved, either by joining clusters if the singular value relative gap improves (Theorem 5.7), or if not, by doing nothing (Theorem 5.9). Finally, Theorem 5.10 proves that it is not necessary to join more than two clusters. Let us remark that the only case in which Algorithm 2 has to be modified to get (54) is when the hypotheses of Theorem 5.7 are satisfied.

In subsection 5.2 we implement a routine, Algorithm 3.1, that merges pairs of adjacent singular value clusters, previously chosen according to (29), whenever the following conditions are met: (a) both clusters are signed with opposite sign, (b) the singular value relative gap is sufficiently smaller than the eigenvalue relative gap, and (c) the singular value relative gap increases after merging the two clusters. Algorithm 2.2 is then applied to these new clusters and Theorem 5.12 proves that (54) is achieved for the computed bases of the invariant subspaces.

Here, as in section 4, only clusters of nonzero singular values will be considered. Apart from the reasons stated in section 4, it should be remarked that a cluster of zero singular values is at the same time a cluster of zero eigenvalues, and both its eigenvalue and singular value relative gaps are equal to 1. Thus for such a cluster an error bound $O(\epsilon\kappa(R')\max(\kappa(X),\kappa(Y)))$ holds, and this cannot be improved. Moreover, a cluster of zero singular values is as far as possible, in relative distance, from any other cluster, thus joining it to another cluster makes no sense.

5.1. Eigenvalue versus singular value relative gaps. Throughout this section we consider a set of real numbers $\Lambda = \{\lambda_1, \dots, \lambda_n\}$ decreasingly ordered, i.e., $\lambda_1 \geq \cdots \geq \lambda_n$, and the set of their moduli, $\Sigma = \{\sigma_1, \dots, \sigma_n\}$, also in decreasing order, i.e., $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. Let Π be the index permutation such that $\sigma_i = |\lambda_{\Pi(i)}|$. Whenever we consider a subset $\Sigma_1 = \{\sigma_{i+1}, \sigma_{i+2}, \dots, \sigma_{i+d_1}\}$ of Σ we will denote by $\Lambda_1 = \{\lambda_{\Pi(i+1)}, \dots, \lambda_{\Pi(i+d_1)}\}$ the corresponding subset of Λ; moreover, we will call Λ_1^+ (resp., Λ_1^-) the set of positive (resp., negative) elements of Λ_1. It is worth thinking of Λ and Σ as being, respectively, the set of eigenvalues and singular values of the real symmetric matrix A studied in section 4, but notice that the results in this subsection are proved using only elementary properties of real numbers, without any reference to spectral properties. Thus, the proofs of the theorems in this subsection are all elementary but sometimes long and involved, mainly due to dealing with clusters containing more than one element. This is why most of the proofs have been omitted. The proof of Theorem 5.7, one of the more intricate results in the section, is included in a final appendix, in order to give an idea of the techniques employed. The remaining proofs are similar, and those of a nonelementary character may be found in [10, Appendix B].

Our definitions of relative gaps (see (3) and (9)) are convenient and appealing in numerical analysis, but the lack of symmetry in relative errors of the type $|\sigma_j - \sigma_i|/\sigma_i$ is unpleasant from a mathematical point of view and complicates somewhat the statement of the results (see more on these questions and definitions of true relative mathematical distances in [19, 20]). In this sense, an effort has been made to state the theorems in such a way that they can be directly applied to the clusters chosen by Algorithm 3.1.

We begin with a general definition of cluster.

DEFINITION 5.1. *Let C_l be a real number such that $0 \leq C_l < 1$. The subset $\Sigma_1 = \{\sigma_{i+1}, \sigma_{i+2}, \dots, \sigma_{i+d_1}\}$ of Σ is called a* cluster of tolerance C_l *if*

1. $(\sigma_j - \sigma_{j+1}) \leq C_l\,\sigma_j$ *for* $j = i+1, \ldots, i+d_1-1$,

2. $(\sigma_i - \sigma_{i+1}) > C_l\,\sigma_i$ *and* $(\sigma_{i+d_1} - \sigma_{i+d_1+1}) > C_l\,\sigma_{i+d_1}$, *whenever all the indices belong to* $\{1, 2, \ldots, n\}$*; otherwise the corresponding inequality does not appear in the definition.*

Notice that in the case of a cluster of dimension 1 ($d_1 = 1$) the first condition is empty. Notice also that this definition includes the clusters of singular values chosen in Algorithm 2, according to criterion (29), for $C_l = \epsilon\kappa(R')\max\{\kappa(X), \kappa(Y)\}$. The condition $C_l < 1$ appearing in Definition 5.1 is necessary—otherwise the whole set Σ would always be a trivial cluster, independently of the distribution of its elements.

Now we define relative gaps for subsets of Λ and Σ. For the sake of simplicity we will use only one argument.

DEFINITION 5.2. *Let* Λ_2 *and* Σ_1 *be any subsets of, respectively,* Λ *and* Σ. *We define the following relative gaps for both subsets:*

1.
$$rg(\Lambda_2) = \min_{\substack{\lambda_k \in \Lambda_2 \\ \lambda_q \notin \Lambda_2}} \frac{|\lambda_q - \lambda_k|}{|\lambda_k|}.$$

2.
$$relgap(\Lambda_2) = \min\{rg(\Lambda_2)\,,\, 1\}.$$

3.
$$rg(\Sigma_1) = \min_{\substack{\sigma_k \in \Sigma_1 \\ \sigma_q \notin \Sigma_1}} \frac{|\sigma_q - \sigma_k|}{\sigma_k}.$$

4.
$$relgap(\Sigma_1) = \min\{rg(\Sigma_1)\,,\, 1\}.$$

Given a subset Σ_1 of Σ, the relationship between the $relgap(\Sigma_1)$ appearing in Definition 5.2 and *relgap* as defined by (23) and (21) is

$$relgap(\Sigma_1) = relgap(\Sigma_{\bar{1}}, \Sigma_1), \tag{55}$$

where the notation introduced in (31) has been used. Similar comments apply to rg defined in (21) and rg defined above. Although $relgap(\Sigma_1, \Sigma_{\bar{1}})$ is the relative gap appearing in the error analysis of section 4, we have found it simpler, from both theoretical and computational points of view, to deal with $relgap(\Sigma_i)$, which has the elements of the cluster being analyzed in the denominators of the relative errors.[6] Both choices are equivalent, as shown in (24) and, on the other hand, it is possible to reformulate Theorem 2.3 using $relgap(\Sigma_i)$.

The error bounds for invariant subspaces computed using the J-orthogonal algorithm and Algorithm 1 are controlled by the relative gaps *relgap*, of eigenvalues and singular values, respectively, in the previous definition (see Theorem 4.7 and [22, p. 7]). However, in the following it is simpler and more general to use the relative gaps rg. At the end of this section it will be shown that theorems obtained for rg easily imply results for *relgap*.

[6] Notice that notation similar to Definition 5.2 has already been used in the introduction (see (3) and (9)).

We start with this simple lemma.

LEMMA 5.3. *Let* $\Sigma_1 = \{\sigma_{i+1}, \sigma_{i+2}, \ldots, \sigma_{i+d_1}\}$ *be a subset of consecutive elements of* Σ. *Then*

$$rg(\Sigma_1) = \min\left\{\frac{\sigma_i - \sigma_{i+1}}{\sigma_{i+1}}, \frac{\sigma_{i+d_1} - \sigma_{i+d_1+1}}{\sigma_{i+d_1}}\right\},$$

where if the index i *or* $i+d_1+1$ *does not belong to* $\{1, \ldots, n\}$ *the corresponding term does not appear in the minimum.*

This lemma allows a natural definition of the *closest cluster to* Σ_1 *in the relative sense.*

DEFINITION 5.4. *Let* $\Sigma_1 = \{\sigma_{i+1}, \sigma_{i+2}, \ldots, \sigma_{i+d_1}\}$ *be a cluster of tolerance* C_l. *We define its* relative closest cluster $\Sigma_{cl(1)}$ *as the cluster of tolerance* C_l *containing* σ_i *if* $rg(\Sigma_1) = (\sigma_i - \sigma_{i+1})/\sigma_{i+1}$, *or the one containing* σ_{i+d_1+1} *if* $rg(\Sigma_1) = (\sigma_{i+d_1} - \sigma_{i+d_1+1})/\sigma_{i+d_1}$.

It is seen from Lemma 5.3 that, with the possible exception of the cluster containing the smallest singular value, $rg(\Sigma_1) \leq 1$ and then $rg(\Sigma_1) = relgap(\Sigma_1)$. Obviously the last equality also holds whenever $rg(\Sigma_1) < 1$, a condition appearing frequently in the results of this section.

Our first result deals with the case of clusters containing singular values corresponding to positive and negative eigenvalues. This theorem shows that in this case the singular value relative gap of the cluster is not worse, up to a moderate constant, than an eigenvalue relative gap. Thus for clusters of singular values of this kind (54) holds, and it is not necessary to join them to any other cluster.

THEOREM 5.5. *Let* Σ_1 *be a cluster of singular values of tolerance* C_l *with* d_1 *elements such that* $(d_1 - 1)C_l < 1$, *and assume that* Λ_1 *contains both positive and negative elements. Then*

$$\min\{rg(\Lambda_1^+)\,,\, rg(\Lambda_1^-)\} \;\leq\; \frac{1}{1-(d_1-1)\,C_l}\left(1 + \frac{(d_1-1)\,C_l}{rg(\Sigma_1)}\right) rg(\Sigma_1).$$

Some remarks about the bound in the previous theorem are in order: the assumption $(d_1 - 1)\,C_l < 1$ is fulfilled for clusters of any size if we demand $C_l < 1/n$; this is really very mild because the clusters are chosen in practice according to (29) with $C = 1$, i.e., $C_l = \epsilon\kappa(R')\max(\kappa(X), \kappa(Y))$, which is smaller than $1/n$ for moderate values of $\max(\kappa(X), \kappa(Y))$. This has led us to set in the numerical experiments

$$C_l = \min\{\epsilon\kappa(R')\max(\kappa(X), \kappa(Y)), 1/n\}. \tag{56}$$

With this choice the factor $1/(1 - (d_1 - 1)\,C_l)$ is always less than n, but it is just a little greater than 1 when $C_l \approx \epsilon$. The presence of the ratio $C_l/rg(\Sigma_1)$ may look odd because we are bounding precisely the quotient $\min\{rg(\Lambda_1^+)\,,\, rg(\Lambda_1^-)\}/rg(\Sigma_1)$; however, notice that Definition 5.1 and Lemma 5.3 imply

$$C_l < rg(\Sigma_1) \quad \text{and} \quad C_l < relgap(\Sigma_1). \tag{57}$$

The ratio $C_l/rg(\Sigma_1)$ is kept in the bound because $C_l \ll rg(\Sigma_1)$ may often happen. It is convenient to bear in mind that these remarks also hold for the bounds appearing in the next theorems of this section. Notice also that all bounds are greatly simplified in the case of one-dimensional clusters.

Now we consider a signed cluster whose relative closest cluster has at least one singular value corresponding to an eigenvalue with the same sign. In this situation, the

next theorem shows that the singular value relative gap is equivalent to the eigenvalue relative gap up to a moderate constant.

THEOREM 5.6. *Let Σ_1 be a cluster of singular values and Σ_2 its relative closest cluster having d_2 elements, both of tolerance C_l. Let all the elements of Λ_1 have the same sign and at least one element of Λ_2 have the same sign as those of Λ_1. If $(d_2-1)C_l < 1$, then*

$$rg(\Lambda_1) \leq \left(1 + \frac{2}{1-(d_2-1)C_l}\,\frac{(d_2-1)\,C_l}{relgap(\Sigma_1)}\right) rg(\Sigma_1).$$

Theorems 5.5 and 5.6 guarantee that, in order to obtain (54) for all the singular value clusters, we need only deal with signed clusters whose relative closest cluster is oppositely signed. This will be the setting for the rest of the section. The following theorem proves that under mild conditions joining clusters of this kind leads to (54).

THEOREM 5.7. *Let Σ_1 be a cluster of d_1 elements and Σ_2 its relative closest cluster, having d_2 elements, both of tolerance C_l. Suppose that all the elements of Λ_1 have the same sign and all the elements of Λ_2 have the opposite sign. Moreover, assume that $(d-1)C_l < 1$, where $d = \max\{d_1, d_2\}$. If $rg(\Sigma_1) < t < 1$ and*

$$rg(\Sigma_1 \cup \Sigma_2) > \min\{rg(\Sigma_1), rg(\Sigma_2)\}, \tag{58}$$

then

$$\min\{rg(\Lambda_1)\,,\, rg(\Lambda_2)\} \leq \frac{1}{1-t}\left(1 + \frac{1}{1-(d-1)\,C_l} + \frac{1}{1-(d-1)\,C_l}\,\frac{(d-1)\,C_l}{rg(\Sigma_1 \cup \Sigma_2)}\right) rg(\Sigma_1 \cup \Sigma_2).$$

The assumption $rg(\Sigma_1) < t < 1$ means that only singular value clusters whose relative gaps are small enough need to be joined to other clusters in order to obtain (54). In practice we have set $t = relgap(\Lambda_1)/2$. Therefore, if $rg(\Sigma_1) \geq t$, the bound in Theorem 4.7 leads trivially to (54). The assumption (58), $rg(\Sigma_1 \cup \Sigma_2) > \min\{rg(\Sigma_1), rg(\Sigma_2)\}$, is imposed to guarantee that by joining clusters Σ_1 and Σ_2 when computing bases of invariant subspaces some improvement is achieved in the bound in Theorem 4.7. In this regard one may wonder what happens with $\max\{rg(\Sigma_1), rg(\Sigma_2)\}$; i.e., how much can the bound (51) worsen for the cluster with the maximum relative gap when Σ_1 and Σ_2 are joined? The next lemma shows that no significant worsening may occur.

LEMMA 5.8. *If both (58) and $rg\{\Sigma_1\} < t < 1$ are fulfilled, then*

$$\max\{rg(\Sigma_1), rg(\Sigma_2)\} < \frac{rg(\Sigma_1 \cup \Sigma_2)}{1-t}.$$

Notice that the difference between the maximum and the minimum values of $\{rg(\Sigma_1), rg(\Sigma_2)\}$ is in this case again a consequence of the lack of symmetry of the relative error.

In order to obtain (54) for all the clusters, we have to prove that if Σ_1 and its relative closest cluster Σ_2, defined as in Theorem 5.7, do not fulfill (58), they will not be joined because Σ_1 has a singular value relative gap not worse, up to a moderate constant, than either its eigenvalue relative gap or the eigenvalue relative gap of Σ_2. Proving this is the goal of the next theorem.

THEOREM 5.9. *Let Σ_1 be a cluster of d_1 elements and Σ_2 its relative closest cluster, having d_2 elements, both of tolerance C_l. Suppose that all the elements of Λ_1 have the same sign and all the elements of Λ_2 have the opposite sign. Moreover, assume that $(d-1)C_l < 1$, where $d = \max\{d_1, d_2\}$. If $rg(\Sigma_1) < t < 1$ and*

$$rg(\Sigma_1 \cup \Sigma_2) = \min\{rg(\Sigma_1), rg(\Sigma_2)\}, \tag{59}$$

then

$$\min\{rg(\Lambda_1)\,,\, rg(\Lambda_2)\} \leq \frac{1}{1-t}\left(1 + \frac{1}{1-(d-1)\,C_l} + \frac{1}{1-(d-1)\,C_l}\,\frac{(d-1)\,C_l}{rg(\Sigma_1)}\right) rg(\Sigma_1).$$

Observe that hypothesis (59) is simply the negation of (58) because we always have $rg(\Sigma_1 \cup \Sigma_2) \geq \min\{rg(\Sigma_1), rg(\Sigma_2)\}$.

Although similar, the bounds appearing in Theorems 5.7 and 5.9 are different in the following sense. While in Theorem 5.7 $\min\{rg(\Lambda_1)\,,\, rg(\Lambda_2)\} \approx rg(\Sigma_1 \cup \Sigma_2)$ always holds, in Theorem 5.9 $\min\{rg(\Lambda_1)\,,\, rg(\Lambda_2)\} \ll rg(\Sigma_1)$ might occur. Thus the error bounds obtained by replacing in (51) $rg(\Sigma_1)$ with $\min\{rg(\Lambda_1)\,,\, rg(\Lambda_2)\}$ may be pessimistic in the conditions of Theorem 5.9.

Our last result shows that in order to obtain (54), unions of more than two clusters are not necessary. In the following theorem three clusters are considered. Two of them satisfy the assumptions of Theorem 5.7, and the third cluster may be a candidate for joining the others. In this situation it will be proved that the relative singular value gap for the third cluster is equivalent, up to a moderate constant, to its eigenvalue relative gap.

THEOREM 5.10. *Let Σ_1 and Σ_2 be clusters satisfying the hypotheses of Theorem 5.7. Let Σ_3 be another cluster, of tolerance C_l, with all the elements of Λ_3 of the same sign and $rg(\Sigma_3) < t_3 < 1$. If Σ_1 (resp., Σ_2) is the relative closest cluster to Σ_3, and all the elements of Λ_3 have sign opposite to those of Λ_1 (resp., Λ_2), then*

$$rg(\Lambda_3) \;\leq\; \left(1 + \frac{1}{(1-t)(1-t_3)}\,\frac{1}{1-(d-1)\,C_l} + \frac{1+t_3}{1-(d-1)\,C_l}\,\frac{(d-1)\,C_l}{rg(\Sigma_3)}\right) rg(\Sigma_3).$$

As announced after Definition 5.2, all the bounds appearing in this section remain true if every rg is replaced by the corresponding *relgap*. This is easily understood as follows: the left-hand sides of the inequalities decrease if the rg's are replaced by the *relgap*'s, and the new left-hand sides are smaller than or equal to 1. The factors that multiply the rg's appearing in the right-hand sides are all greater than or equal to 1 and increase when quotients of the kind C_l/rg are replaced by $C_l/relgap$. Thus the left-hand sides are bounded simultaneously by 1 and by some factor greater than or equal to 1 times the corresponding rg. Then they are bounded by the factor times the *relgap*. Also notice that for testing the assumptions in the results in this section, it is equivalent to use rg's or *relgap*'s. First, it is trivial to see that $rg(\Sigma_1) < t < 1$ if and only if $relgap(\Sigma_1) < t < 1$. Second, in testing the condition (58), the following elementary lemma holds.

LEMMA 5.11. *Let*

$$\Sigma_1 = \{\sigma_{i+1}, \sigma_{i+2}, \ldots, \sigma_{i+d_1}\}, \quad \Sigma_2 = \{\sigma_{i+d_1+1}, \sigma_{i+d_1+2}, \ldots, \sigma_{i+d_1+d_2}\}$$

be any pair of consecutive clusters of nonzero singular values of tolerance C_l. Then

1. $rg(\Sigma_1 \cup \Sigma_2) = \min\{rg(\Sigma_1), rg(\Sigma_2)\}$ *if and only if* $relgap(\Sigma_1 \cup \Sigma_2) = \min\{relgap(\Sigma_1), relgap(\Sigma_2)\}$.

2. $rg(\Sigma_1 \cup \Sigma_2) > \min\{rg(\Sigma_1), rg(\Sigma_2)\}$ *if and only if* $relgap(\Sigma_1 \cup \Sigma_2) > \min\{relgap(\Sigma_1), relgap(\Sigma_2)\}$.

The key to proving this simple lemma is that $rg(\Sigma_1) \leq (\sigma_{i+d_1} - \sigma_{i+d_1+1})/\sigma_{i+d_1} < 1$; thus the 1 appearing in the *relgap*'s does not play any role. Taking into account the facts that $rg(\Sigma_1 \cup \Sigma_2) \geq \min\{rg(\Sigma_1), rg(\Sigma_2)\}$ and $relgap(\Sigma_1 \cup \Sigma_2) \geq \min\{relgap(\Sigma_1), relgap(\Sigma_2)\}$, statements 1 and 2 in the previous lemma are equivalent.

The final consequence of this section is that in order to get (54) only clusters fulfilling the hypotheses of Theorem 5.7 must be joined. Once a pair of clusters of this kind are joined, they can be disregarded in any other union processes as shown by Theorem 5.10. Otherwise, the rest of the results prove that union of clusters of different kinds is not needed. In the next subsection the task of developing a routine that selects and joins clusters according to this criterion will be undertaken.

5.2. Choosing a new set of clusters. Now we will present a routine for step 3 of Algorithm 3. Given a set of clusters as input, selected according to (29), a new set of clusters will come out according to the logic of the theorems in section 5.1; i.e., clusters will be joined only if the hypotheses of Theorem 5.7 are satisfied. All clusters of singular values appearing in the following algorithm are assumed to contain consecutive singular values. Moreover, we order the clusters $\{\Sigma_i\}$ in such a way that any singular value in Σ_i is smaller than any singular value in Σ_{i-1}.

ALGORITHM 3.1.

Input: Eigenvalues Λ; Clusters $\{\Sigma_i\}_{i=1}^{k}$; tolgap: parameter smaller than 1.
Output: New set of clusters: $\{\Sigma_i\}_{i=1}^{q}$ with $q \leq k$.
Notation: Λ_i denotes the set of eigenvalues whose absolute values are the elements of Σ_i.

1. $q = k$
2. for i=1:k
   $qrg(i) = \frac{relgap(\Sigma_i)}{relgap(\Lambda_i)}$
   if($\lambda_j > 0 \quad \forall\, \lambda_j \in \Sigma_i$) then
   $sign(\Sigma_i) = +1$
   elseif($\lambda_j < 0 \quad \forall\, \lambda_j \in \Sigma_i$)
   $sign(\Sigma_i) = -1$
   else
   $sign(\Sigma_i) = 0$
   $qrg(i) = 2$
   endif
   endfor
3. $qrgmin = \min_{1 \leq i \leq q} qrg(i) \equiv qrg(i_c)$
4. while $qrgmin <$ tolgap
   determine the relative closest[7] cluster to Σ_{i_c} according to Definition 5.4. Assume that it is Σ_{i_c+1}.
   if ($sign(\Sigma_{i_c}) * sign(\Sigma_{i_c+1}) = -1$) and
   ($relgap(\Sigma_{i_c} \cup \Sigma_{i_c+1}) > \min\{relgap(\Sigma_{i_c}), relgap(\Sigma_{i_c+1})\}$) then
   $q = q - 1$
   $relgap(\Sigma_{i_c}) = relgap(\Sigma_{i_c} \cup \Sigma_{i_c+1})$

[7] The same can be done if Σ_{i_c-1} is the relative closest cluster to Σ_{i_c}.

$sign(\Sigma_{i_c}) = 0$
$\Sigma_{i_c} = \Sigma_{i_c} \cup \Sigma_{i_c+1}$
`for` $j = i_c + 1 : q$
$\Sigma_j = \Sigma_{j+1}$
$relgap(\Sigma_j) = relgap(\Sigma_{j+1})$
$sign(\Sigma_j) = sign(\Sigma_{j+1})$
`endfor`
`endif`
$qrg(i_c) = 2$
$qrgmin = \min_{1 \le i \le q} qrg(i) \equiv qrg(i_c)$
`5. endwhile`

In practice we have set `tolgap` $= 1/2$, but other values are admissible. This choice leads to values $t = (relgap(\widehat{\Lambda}_i)/2) \le 1/2$ for the parameters t appearing in Theorems 5.7, 5.9, and 5.10.

For the new set of clusters selected by Algorithm 3.1, the error in the corresponding bases of invariant subspaces computed by Algorithm 2.2 *is given by Theorem* 4.7 *using the new singular value relative gaps, and these are the sharpest bounds we have for Algorithm* 3. Nevertheless, in the next theorem we will use the theorems in the previous subsection to give an upper bound for the inverse of the new singular value relative gaps in (51) in terms of inverses of the eigenvalue relative gaps. Therefore this theorem gives a precise statement of (54).

THEOREM 5.12. *Let A be a $n \times n$ real symmetric matrix of rank r for which it is possible to compute an RRD fulfilling* (10). *Let $\widehat{\Sigma}$ be the singular values of A computed using steps* 1–2 *of Algorithm* 1. *Let $\widehat{\Sigma}_i$, $i = 1, \ldots, q$, be the set of clusters of nonzero computed singular values of A selected by* step 3 *of Algorithm* 3, $\widehat{\Lambda}_i = \widehat{\Lambda}_i^+ \cup \widehat{\Lambda}_i^-$, $i = 1, \ldots, q$, *the corresponding set of clusters of eigenvalues, and $\widehat{Q}_i = [\widehat{Q}_i^+ \ \widehat{Q}_i^-]$, $i = 1, \ldots, q$, the matrices computed by* step 4 *of Algorithm* 3. *Let Σ_i (resp., Λ_i) , $i = 1, \ldots, q$, be the corresponding clusters of exact singular values (resp., eigenvalues).*

1. *If neither $\widehat{\Lambda}_i^+$ nor $\widehat{\Lambda}_i^-$ are empty, then there exist matrices Q_i^+ and Q_i^-, whose columns form orthonormal bases of the invariant subspaces of A corresponding, respectively, to the positive and negative eigenvalues of Λ_i, such that*

$$\|\widehat{Q}_i^+ - Q_i^+\|_F \le \frac{O(\epsilon \kappa(R') \max(\kappa(X), \kappa(Y)))}{\min\{relgap(\widehat{\Lambda}_i^+), relgap(\widehat{\Lambda}_i^-)\}}, \tag{60}$$

with a similar bound for $\|\widehat{Q}_i^- - Q_i^-\|_F$.

2. *If all the elements of $\widehat{\Lambda}_i$ have the same sign and $relgap(\widehat{\Sigma}_i) \ge$ `tolgap` $*$ $relgap(\widehat{\Lambda}_i)$, then there exists a matrix Q_i, whose columns form an orthonormal basis of the invariant subspace of A corresponding to the eigenvalues in Λ_i, such that*

$$\|\widehat{Q}_i - Q_i\|_F \le \frac{O(\epsilon \kappa(R') \max(\kappa(X), \kappa(Y)))}{relgap(\widehat{\Lambda}_i)}. \tag{61}$$

3. *If all elements of $\widehat{\Lambda}_i$ have the same sign, $relgap(\widehat{\Sigma}_i) <$ `tolgap` $* relgap(\widehat{\Lambda}_i)$, and the relative closest cluster $\widehat{\Sigma}_{cl(i)}$ to $\widehat{\Sigma}_i$ has all the corresponding eigenvalues with the opposite sign, then there exists a matrix Q_i, whose columns form an orthonormal*

basis of the invariant subspace of A corresponding to the eigenvalues in Λ_i, such that

$$\|\widehat{Q}_i - Q_i\|_F \leq \frac{O(\epsilon\kappa(R')\max(\kappa(X),\kappa(Y)))}{\min\{relgap(\widehat{\Lambda}_i), relgap(\widehat{\Lambda}_{cl(i)})\}}. \tag{62}$$

4. *If all elements of $\widehat{\Lambda}_i$ have the same sign, $relgap(\widehat{\Sigma}_i) < \texttt{tolgap} * relgap(\widehat{\Lambda}_i)$, and the relative closest cluster to $\widehat{\Sigma}_i$ does not have all the corresponding eigenvalues with the opposite sign, then there exists a matrix Q_i, whose columns form an orthonormal basis of the invariant subspace of A corresponding to the eigenvalues in Λ_i, such that*

$$\|\widehat{Q}_i - Q_i\|_F \leq \frac{O(\epsilon\kappa(R')\max(\kappa(X),\kappa(Y)))}{relgap(\widehat{\Lambda}_i)}. \tag{63}$$

Furthermore, let $\widehat{Q} = [\widehat{Q}_1^+ \ \widehat{Q}_1^- \ldots \widehat{Q}_q^+ \ \widehat{Q}_q^-]$ be the $n \times r$ matrix whose columns are the bases of all considered invariant subspaces of A computed using step 4 *of Algorithm* 3. *Then there exists an $n \times r$ matrix B with exact orthonormal columns such that*

$$\|\widehat{Q} - B\|_F = O(\epsilon). \tag{64}$$

Proof. The proof follows from Theorem 4.7 applied to the output clusters of Algorithm 3.1 (step 3 of Algorithm 3) and the theorems on gaps in section 5.1 with $C_l = \epsilon\kappa(R')\max(\kappa(X),\kappa(Y))$. As remarked after Theorem 5.10, *relgap*'s instead of *rg*'s can be used in these theorems.

We begin by replacing $relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})$ with $relgap(\widehat{\Sigma}_{\bar{i}}, \Sigma_i)$ in the bound (51). This does not significantly change the bound due to (24). Moreover, we assume that $relgap(\widehat{\Sigma}_{\bar{i}}, \Sigma_i) \approx relgap(\widehat{\Sigma}_{\bar{i}}, \widehat{\Sigma}_i)$. This is a fair assumption whenever steps 1–2 of Algorithm 1 compute singular values with high relative accuracy. Thus (55) allows us to apply (51), with $relgap(\Sigma_i, \widehat{\Sigma}_{\bar{i}})$ replaced by $relgap(\widehat{\Sigma}_i)$, to the clusters selected by Algorithm 3.1.

Consider a cluster $\widehat{\Sigma}_{i_c}$ of singular values corresponding to the quantity *qrgmin* in Algorithm 3.1. This cluster is joined to its relative closest cluster if and only if the following three conditions are simultaneously fulfilled:

(c1) $qrg(i_c) = \frac{relgap(\widehat{\Sigma}_{i_c})}{relgap(\widehat{\Lambda}_{i_c})} < \texttt{tolgap} < 1$.

(c2) $sign(\widehat{\Sigma}_{cl(i_c)}) * sign(\widehat{\Sigma}_{i_c}) = -1$, where $\widehat{\Sigma}_{cl(i_c)}$ is the closest cluster to $\widehat{\Sigma}_{i_c}$.

(c3) $relgap(\widehat{\Sigma}_{i_c} \cup \widehat{\Sigma}_{cl(i_c)}) > \min\{relgap(\widehat{\Sigma}_{i_c}), relgap(\widehat{\Sigma}_{cl(i_c)})\}$.

If all three conditions (c1), (c2), and (c3) are fulfilled, Algorithm 3.1 joins $\widehat{\Sigma}_{i_c}$ and $\widehat{\Sigma}_{cl(i_c)}$ in a new output cluster $\widehat{\Sigma}_{i_c} \cup \widehat{\Sigma}_{cl(i_c)}$. In this case Theorem 5.7 applies with $t = \texttt{tolgap} * relgap(\widehat{\Lambda}_{i_c})$. This together with (51) yields (60) for the eigenvectors corresponding to the new output cluster.

Now, suppose that at least one of the three conditions is not satisfied. Suppose first that (c1) is satisfied, which implies $sign(\widehat{\Sigma}_{i_c}) \neq 0$; otherwise $qrgmin = 2$. If (c2) is not verified and the closest cluster to $\widehat{\Sigma}_{i_c}$ is an input cluster, Theorem 5.6 can be applied to the bound (51) to obtain (63); on the other hand, if (c2) is not verified and the closest cluster is a new output cluster, (63) is achieved by using Theorem 5.6 or 5.10. If (c2) is verified and (c3) is also verified, we are in the previously studied case of joining clusters. If (c2) is verified and (c3) is not verified, Theorem 5.9 can be applied to (51) to yield (62).

Suppose from now on that (c1) is not satisfied. Then, Algorithm 3.1 stops and all the clusters existing at that moment verify

$$qrg(i) \geq \texttt{tolgap}, \quad i = 1, \ldots, q.$$

If $sign(\widehat{\Sigma}_i) = 0$, this is either because $sign(\widehat{\Sigma}_i) = 0$ on input or because $\widehat{\Sigma}_i$ is a new output cluster, i.e., union of two input clusters. Anyway, Theorem 5.5 or 5.7 leads to (60) by using (51). If $sign(\widehat{\Sigma}_i) \neq 0$ and $qrg(i) = 2$, then $\widehat{\Sigma}_i$ already has been analyzed inside the `while` loop and, according to the previous paragraph, either (62) or (63) is satisfied. If $sign(\widehat{\Sigma}_i) \neq 0$ but $\texttt{tolgap} \leq relgap(\widehat{\Sigma}_i)/relgap(\widehat{\Lambda}_i) \leq 1$, then (51) implies (61) at the cost of an additional factor $1/\texttt{tolgap}$. With this, all the possible cases on the decision tree for the conditions (c1), (c2), and (c3) have been studied. The proof of (64) is as in Theorem 4.7. □

We finish this section with two important remarks.

Remark 1. The *eigenvalue* clusters treated in the last theorem are exactly the same as the ones corresponding to the singular value clusters chosen according to (29). This is because Algorithm 3.1 only joins oppositely signed clusters and Algorithm 2.2 computes the bases separately.

Remark 2. The bounds in Theorem 5.12 have been obtained in two stages: first, applying Theorem 4.7 to the new set of clusters produces a bound depending on singular value relative gaps. Then, this bound is majorized by other ones, depending on certain eigenvalue relative gaps. This second stage never worsens significantly the first bound, except in case 3 of Theorem 5.12. Thus, the bound (62) may be pessimistic, because the quantity $\min\{relgap(\widehat{\Lambda}_i), relgap(\widehat{\Lambda}_{cl(i)})\}$ might be much smaller than $relgap(\widehat{\Sigma}_i)$. However, recall that the sharpest bound for Algorithm 3 is of the order of $\epsilon\kappa(R')\max(\kappa(X), \kappa(Y))/relgap(\widehat{\Sigma}_i)$.

6. Numerical experiments. In this section we present results of two types of numerical experiments. First, we test Algorithm 3, the third step of Algorithm 1, in a setting where the errors for steps 1 and 2 of Algorithm 1 are controlled. A second kind of experiment tests the entire Algorithm 1, including the computation of the RRD in two different ways, as either a symmetric RRD of the form $A = XDX^T$ or a nonsymmetric RRD of the form $A = XDY^T$. We also include experiments for Algorithm 1 with Algorithm 2 in step 3. Thus the reader can check that Algorithm 3 really improves the accuracy of the eigenvectors in the few cases in which Algorithm 2 delivers eigenvectors with large errors. When needed, we will distinguish between the two versions of Algorithm 1: the version with Algorithm 2 in step 3 will be called `SSVDO`, and the one with Algorithm 3 will be called simply `SSVD`. Besides, a first subsection describes some practical details of the implementation of the three steps of Algorithm 1.

As will be seen from the experiments in subsection 6.2, Algorithm 1 behaves as predicted by the error analysis in sections 4 and 5 and compares well in both the sense of accuracy and of speed with the J-orthogonal algorithm.

6.1. Implementation of Algorithm 1.

1. The RRD of the matrix A in step 1 of Algorithm 1 has been done in the following two ways:

- symmetric RRD, $A = XDX^T$, using a modification of the symmetric indefinite Bunch and Parlett (BP) decomposition [3]; more specifically, we have used an adapted version of the routine `SGJGT` in [22].

- a nonsymmetric RRD, $A = XDY^T$, by means of an LU factorization with complete pivoting (Gaussian elimination with complete pivoting (GECP)). We have used a modification of the LAPACK procedure `SGETF2`.

2. The SVD in step 2 of Algorithm 1 has been done using Algorithm 3.1 of [6]. Only LAPACK and BLAS routines have been used, as in [6], except for the one-sided Jacobi code in which we have used a routine developed by Z. Drmač according to the ideas in [12]. The implementation of the procedure (called `SGEPSV` in [6] in single precision) has the following steps.

ALGORITHM 4. `(SGEPSV)` (ALGORITHM 3.1 IN [6].)
`Input:` $X, D, Y : A = XDY^T$.
`Output:` $U, \Sigma, V : A = U\Sigma V^T$.
1. `QR factorization with column pivoting of` XD, $XDP = QR$; $A = QRP^TY^T$
   `LAPACK Routine: SGEQPF`
2. `Multiply to get` $W = R(YP)^T$; $A = QW$
   `BLAS Routine: STRMM`
3. `SVD of W with one-sided Jacobi;` $W = \overline{U}\Sigma V^T$; $A = Q\overline{U}\Sigma V^T$
   `Routine: S_SGESVDJ developed by Z. Drmač [12]`
4. `Multiply` $U = Q\overline{U}$; $A = U\Sigma V^T$
   `LAPACK Routine: SORMQR`

Two versions of this algorithm have been used, depending on whether right-Jacobi (right multiplication on W by Jacobi plane rotations) or left-Jacobi (right multiplication on W^T by Jacobi plane rotations) is employed in the one-sided Jacobi step 3 of Algorithm 4 in [6]. The left-Jacobi version has the advantage of speeding up the convergence. Although the error bounds for this version are weaker than for the other version (see [11] or [10, Appendix A]), no significant difference in accuracy has ever been observed in practice. Our experiments confirm this.

In any case the routine that has been used computes one of the singular vector matrices by a product of Jacobi plane rotations. There exist much faster, equally accurate, versions of one-sided Jacobi algorithms which do not accumulate rotations [14], and which could also be used. Nevertheless, with the present implementation the timing statistics of Algorithm 1 are comparable to the J-orthogonal algorithm (see the timing data in the last paragraph of Experiment 2 in subsection 6.2 below).

3. Algorithm 2 in step 3 of Algorithm 1 has been implemented as described in subsection 3.3. Algorithm 3, the final version of step 3 in Algorithm 1, has been implemented as described in section 5. Some additional specific details are the following:

(i) Recall that steps 1 and 2 are the same in both Algorithms 2 and 3, and therefore the eigenvalues computed by both algorithms are the same.

(ii) The choice of clusters in step 1 of Algorithms 2 and 3 has been done using (29) by taking $C = 1$ and using the $O(n^2)$ estimator LAPACK routine `STRCON` to estimate $\kappa(R')$, or $\kappa(X)$, $\kappa(Y)$, when starting from a nonfactorized matrix. When generating matrices in RRD form $A = XDX^T$, some matrices X producing values of $\epsilon\kappa(R')\kappa(X)$ larger than 1 have appeared. This means that the SVD routine, Algorithm 4, guarantees no significant digits of precision in the computation of the singular values. Moreover, using (29) produces in this case that all singular values are contained in just one cluster. Our discussion after Theorem 5.5 has led us to establish in practice the criterion to include two contiguous singular values σ_j, σ_{j+1} in the same cluster

whenever

$$\frac{|\sigma_j - \sigma_{j+1}|}{\sigma_j} \leq \min\{\epsilon\kappa(R')\max\{\kappa(X),\kappa(Y)\},1/n\}. \tag{65}$$

(iii) The product $\Delta_i = V_i^T U_i$ in step 11 of Algorithm 2.2 has been done using the BLAS routine `SGEMM`.

(iv) The diagonalization of $\Delta_i = [W_i^+\, W_i^-]J_i[W_i^+\, W_i^-]^T$ (step 12 of Algorithm 2.2) has been done using the LAPACK routine `SSYEV` applied only to the triangular upper half of the matrix, as assumed in Lemma 4.5. Finally, the eigenvector matrices $Q_i^{\pm} = V_i W_i^{\pm}$ (step 13 of Algorithm 2.2) are obtained using the BLAS multiplication routine `SGEMM`.

(v) In all the experiments the value for the parameter `tolgap` appearing in Algorithm 3.1 has been set to `tolgap = 1/2`.

6.2. Numerical results. The following experiments were done using an AMD K7 ATHLON processor with IEEE arithmetic, and the routines were implemented with Fortran PowerStation 4.0 from Microsoft. All numerical experiments in this section have been done with nonsingular matrices, although as pointed out in sections 3 and 4, Algorithm 1 also can be applied to rank-deficient matrices.

In the first experiment we start from matrices already in factorized RRD form $A = XDX^T$, directly generating the matrices X and D. This has helped us to focus on the accuracy of step 3 in Algorithm 1 since, given the RRD, the work by Demmel et al. in [6] allows us to control the error in step 2 of Algorithm 1.

In the second group of experiments, two different kinds of nonfactorized test matrices have been generated: graded matrices and matrices specifically designed in [22] to guarantee a good performance of the J-orthogonal algorithm. The reason for choosing graded matrices is that it is known, under the conditions given in [6, section 4], that an accurate RRD, in the sense of (10), can be computed using a *plain implementation* of GECP. For the rest of the classes of matrices treated in [6, pp. 26–27], special implementations of GECP are needed to get the desired accuracy, and it is unfair to compare in these cases Algorithm 1 with the J-orthogonal algorithm, since at present no special implementations of the symmetric indefinite factorization are known to guarantee the accuracy. The reason for choosing the matrices designed in [22] is to compare Algorithm 1 and the J-orthogonal algorithm on matrices where the accuracy of the J-orthogonal algorithm of the latter is guaranteed.

To test Algorithm 1 we have used as reference the eigenvalues and eigenvectors computed by the routine `DSYEVJ`, developed by I. Slapničar, that implements the implicit one-sided J-orthogonal algorithm[8] [22] in double precision ($\epsilon = \epsilon_D \approx 1.11 \times 10^{-16}$). From now on these eigenvalues and eigenvectors are denoted, respectively, simply by λ_i and q_i. These are compared with the eigenvalues and eigenvectors, $\lambda_i^{(S)}$ and $q_i^{(S)}$, computed in single precision ($\epsilon = \epsilon_s \approx 5.96 \times 10^{-8}$) by the following routines: `SSVDO` (Algorithm 1, using Algorithm 2 in step 3), `SSVD` (Algorithm 1, using Algorithm 3 in step 3), `SSYEVJ` (J-orthogonal algorithm, denoted simply by J-O in the tables and figure), and, only when we start from a full (not already in rank-

[8] `DSYEVJ` is a driver routine formed by two routines that implement the two steps of the J-orthogonal algorithm: subroutine `DGJGT` (symmetric indefinite decomposition with complete pivoting) and subroutine `DJGJF` (implicit J-orthogonal Jacobi method with the same stopping criterion as one-sided Jacobi). `DSYEVJ` has been used when starting with the full matrix A. When starting from a factorized matrix $A = XDX^T$ only the subroutine `DJGJF` has been used. Similar remarks apply to the single precision driver routine `SSYEVJ`.

revealing form) matrix A, `SJAC` (standard Jacobi algorithm with the new stopping criterion introduced in [7, p. 1206] with `tol` $= \epsilon_s$) and `SSYEV` (LAPACK driver routine that implements tridiagonalization followed by QR iteration). For these methods the following quantities have been measured for each test matrix:

1. The maximum relative error in the eigenvalues:

$$e_\lambda^{(\mathrm{S})} = \max_i \left| \frac{\lambda_i - \lambda_i^{(\mathrm{S})}}{\lambda_i} \right|. \tag{66}$$

2. A control quantity for eigenvalues:

$$\vartheta^{(\mathrm{S})} = \frac{e_\lambda^{(\mathrm{S})}}{\kappa\, \epsilon_s}, \tag{67}$$

where $\kappa = \kappa(R') \max\{\kappa(X), \kappa(Y)\}$, as in (15). Observe that when referring to symmetric RRDs κ is just $\kappa(R')\kappa(X)$. According to the bound (38), the quantity $\vartheta^{(\mathrm{S})}$ is expected to be $O(1)$ for Algorithm 1. For the J-orthogonal algorithm the error $e_\lambda^{(\mathrm{S})}$ is essentially bounded by $O(\epsilon_s\, \kappa(XD_X))$, where XD_X is the best conditioned column diagonal scaling of matrix X [22]. However, we have checked that $\kappa(X) \approx \kappa(XD_X)$ in our tests. This is due to the fact that the matrices X appearing in our experiments do not have any special structure. Furthermore, the extra factor $\kappa(R')$ in the denominator that we have observed is $O(n)$ in the numerical tests in this section (see also [6, Thm. 3.2]) renders $\vartheta^{(\mathrm{S})}$ inadequate to check how well the bounds for the J-orthogonal algorithm behave, although it is still valid to compare the accuracy of Algorithm 1 and the J-orthogonal algorithm. For the other two considered algorithms, Jacobi and QR, $\vartheta^{(\mathrm{S})}$ is just the maximum error in the eigenvalues normalized in the same way as for both Algorithm 1 and the J-orthogonal algorithm. Similar remarks apply to the eigenvector computations.

3. Corresponding to each cluster of eigenvalues, the sine of the maximum of canonical angles between the subspaces spanned by the computed basis, Q_i, in double precision and the computed basis, $Q_i^{(\mathrm{S})}$, in single precision:

$$E_{\Lambda_i}^{(\mathrm{S})} = \| \sin \Theta(Q_i, Q_i^{(\mathrm{S})}) \|_2. \tag{68}$$

In the case of clusters with one single element we have computed just the Euclidean norm of the difference between the computed eigenvectors in double, q_i, and single $q_i^{(\mathrm{S})}$, precision,

$$e_{q_i}^{(\mathrm{S})} = \| q_i - q_i^{(\mathrm{S})} \|_2. \tag{69}$$

Actually, the quantities $e_{q_i}^{(\mathrm{S})}$ are always computed, even in the presence of clusters of dimension larger than one. We do this in order to check that clusters are only chosen whenever no accuracy can be guaranteed for individual computed eigenvectors.

4. The control quantities for bases of invariant subspaces are

$$\Xi_\Sigma^{(\mathrm{S})} = \max_i \frac{E_{\Lambda_i}^{(\mathrm{S})}\ \mathit{relgap}(\Sigma_i^{(\mathrm{S})})}{\kappa\, \epsilon_s}, \qquad \Xi_\Lambda^{(\mathrm{S})} = \max_i \frac{E_{\Lambda_i}^{(\mathrm{S})}\ \mathit{relgap}(\Lambda_i^{(\mathrm{S})})}{\kappa\, \epsilon_s}, \tag{70}$$

and the corresponding ones for individual eigenvectors are

$$\xi_\sigma^{(\mathrm{S})} = \max_i \frac{\|q_i - q_i^{(\mathrm{S})}\|_2 \;\; relgap(\sigma_i^{(\mathrm{S})})}{\kappa\, \epsilon_s},$$
$$\xi_\lambda^{(\mathrm{S})} = \max_i \frac{\|q_i - q_i^{(\mathrm{S})}\|_2 \;\; relgap(\lambda_i^{(\mathrm{S})})}{\kappa\, \epsilon_s}. \tag{71}$$

According to Theorem 4.7, $\Xi_\Sigma^{(\mathrm{S})}$ and $\xi_\sigma^{(\mathrm{S})}$ are expected to be $O(1)$ for Algorithms `SSVD` and `SSVD0`. Also $\Xi_\Lambda^{(\mathrm{S})}$ and $\xi_\lambda^{(\mathrm{S})}$ are expected to be $O(1)$ for the J-orthogonal algorithm, but not for Algorithms `SSVD` and `SSVD0`, because the accuracy of `SSVD` is governed by Theorem 5.12. However, the quantities $\Xi_\Lambda^{(\mathrm{S})}$ and $\xi_\lambda^{(\mathrm{S})}$ will be computed by `SSVD` and `SSVD0` to check in practice how the `SSVD` algorithm improves the accuracy of `SSVD0` and how it compares with the J-orthogonal algorithm. Notice that the quantities $relgap(\Sigma_i^{(\mathrm{S})})$ correspond either to the set of cluster chosen according to (65) for Algorithm `SSVD0` or to the output clusters of Algorithm 3.1 for Algorithm `SSVD`. The quantities $relgap(\Lambda_i^{(\mathrm{S})})$ are always the same because the clusters for eigenvalues do not change (see the remarks at the end of subsection 5.2). The $relgap$s in (71) are the ones defined in (3) and (9) for any of the algorithms.

For the sake of brevity, values of $\xi_\sigma^{(\mathrm{S})}$ or $\xi_\lambda^{(\mathrm{S})}$ are not shown for routines `SJAC` and `SSYEV`; we simply report that extremely large errors are obtained for these algorithms.

To do our experiments we have generated matrices in single precision in different ways. All the random matrices needed have been generated using the LAPACK routines `SLATM1`, for diagonal matrices, and `SLATMR`, for full matrices. When we have generated matrices with a fixed condition number $\mathcal{K}$, it has been done by producing diagonal matrices with elements of absolute values in the range from 1 to $1/\mathcal{K}$, and after that multiplying by random single precision orthogonal matrices. The distribution of the diagonal elements is controlled by the parameter `MODE` of the routine `SLATM1`: $|$`MODE`$| = 3$, geometrically distributed; $|$`MODE`$| = 4$, arithmetically distributed; `MODE` $= 5$, with logarithms uniformly distributed. If `MODE` is positive (resp., negative) the elements are set in decreasing (resp., increasing) order.

Experiment 1. This experiment is designed to test Algorithms 2 and 3. We have generated $n \times n$ matrices X and D (diagonal), factors of a matrix $A = XDX^T$, as done in [6]. Parameters have been chosen as follows: $\kappa(X) = 10^{[2:1:6]}$; $\kappa(D) = 10^{[2:2:16]}$; $MODE_X = 3, 4, 5$; $MODE_D = \pm 3, \pm 4, 5$. For each set of parameters we have run 20 matrices for $n = 50, 100$ (total 12000 matrices for each n), 2 for $n = 250$ (total 1200 matrices), 2 for $n = 500$ (total 1200 matrices), 1 for $n = 1000$, and only for 2 combinations of the $MODE$s (total 80 matrices).

Figure 6.1 shows the maximum, minimum, and average (over all $MODE$s, samples, and $\kappa(D)$s) of the quantity $log_{10} e_\lambda^{(\mathrm{S})}$, roughly the number of correct digits in the computed eigenvalues, as a function of $\kappa(X)$ for $n = 100$ for Algorithm 1 (`SSVD` or `SSVD0`) and for the J-orthogonal algorithm. The line $\epsilon_s \kappa(X) \kappa(R')$ is plotted as a guide to the eye; the quantity $\kappa(R')$ in this line is really the average of $\kappa(R')$ over all the matrices with that value of $\kappa(X)$. The results confirm the theoretical error bounds for eigenvalues.

Table 6.1 shows the statistical data corresponding to the quantity $\vartheta^{(\mathrm{S})}$. The aim is to check the bound (38) for Algorithm 1 and compare its accuracy against the J-orthogonal algorithm. The most significant data in Table 6.1 appear under the columns labeled "max" where the maximum values of each magnitude (the ones bounded by the error analysis) are shown. In particular, the fact that the quantities

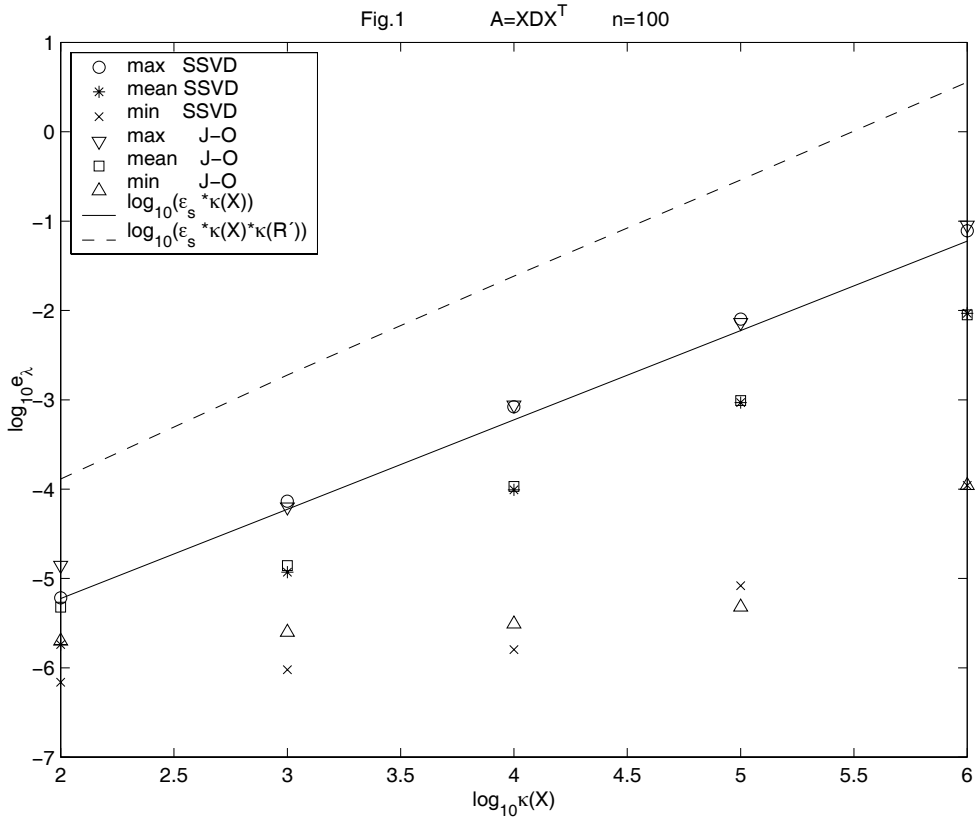

FIG. 6.1. *Experiment* 1. *Maximum relative error for eigenvalues:* $\log_{10} e_\lambda^{(\mathrm{S})}$ *against* $\log_{10} \kappa(X)$.

TABLE 6.1
Experiment 1. *Statistical data for accuracy in eigenvalues:* $\vartheta^{(\mathrm{S})}$.

n	50		100		250		500		1000	
Method	mean	max	mean	max	mean	max	mean	max	mean	max
$\vartheta^{(\mathtt{SSVD})}$	.030	.40	.022	.31	.015	.17	.013	.22	.013	.20
$\vartheta^{(\mathtt{J-O})}$	.041	.58	.037	.44	.039	.47	.044	.63	.050	.65
$\vartheta^{(\mathtt{SVD})}$	.030	.40	.022	.31	.015	.17	.013	.22	.012	.20

in the first row are smaller than 1 confirms that Algorithm 1 satisfies the bound (38). In addition, the third row itself is the control quantity ϑ calculated for the singular values computed in step 2 of Algorithm 1. The comparison of the first and third rows shows that Algorithm 1 never misses a sign and always gives eigenvalues with the same precision as the singular values, except for five matrices of dimension 1000. These cases have $\kappa(X) = 10^6$ and $\epsilon_s \kappa(X)\kappa(R')$ greater than 100. Therefore *whenever* $\epsilon_s \kappa(X)\kappa(R') < 1$ *Algorithm* 1 *has given the eigenvalues with the same precision as the singular values computed by Algorithm* 3.1 *in* [6]. It can be seen, from both Figure 6.1 and Table 6.1, that Algorithm 1 performs for eigenvalues as well (even a little better, especially for small values of $\kappa(X)$) as the J-orthogonal algorithm, with the maximum errors in Algorithm 1 adjusting very well to the predicted behavior $\epsilon\kappa(X)\kappa(R')$. It can be observed also that the data do not depend on n.

TABLE 6.2
Experiment 1. *Statistical data for the number of sweeps.*

n	50		100		250		500		1000	
Method	mean	max	mean	max	mean	max	mean	max	mean	max
Sweeps (SSVD)	5.5	10	6.3	12	7.4	12	8.4	14	9.3	15
Sweeps (J-O)	10.5	20	11.7	22	13.0	22	13.9	24	13.1	24

Moreover, for a significant portion of all the matrices (4144 matrices out of 12000 for $n = 50$; 6693 matrices out of 12000 for $n = 100$; 974 matrices out of 1200 for $n = 250$; 1105 matrices out of 1200 for $n = 500$; 79 matrices out of 80 for $n = 1000$), clusters of singular values of dimension greater than 1, according to criterion (65), have been found, with the maximum dimension of a cluster being 5. The average number of clusters has ranged from almost no clusters for $n = 50$ to approximately 40 clusters for $n = 1000$, with a typical dimension of 2. This shows that criterion (65) chooses clusters which determine perfectly in practice the signs of the eigenvalues. After applying Algorithm 3.1 all the considered matrices have clusters. The average number of clusters in this case is approximately $0.3n$ for all n.

In Table 6.2 we show the statistics for the number of orthogonal Jacobi sweeps for Algorithm SSVD and the number of hyperbolic Jacobi sweeps for the J-orthogonal algorithm. These data correspond to the use of left-Jacobi in step 3 of Algorithm 4. If right-Jacobi is used, the average number of sweeps for Algorithm SSVD is 13.8, with a maximum of 28 for $n = 100$, while the accuracy is the same. For these reasons, we have used in the rest of our experiments the left-handed version of the algorithm. It can be seen that the J-orthogonal algorithm uses more sweeps than Algorithm SSVD: on average, from 5 more for $n = 50$ to almost 4 for $n = 1000$.

Now we focus on the analysis of data both for eigenvectors and for bases of invariant subspaces. Table 6.3 shows the quantities $\Xi_\Sigma^{(S)}$ and $\Xi_\Lambda^{(S)}$ defined in (70) for Algorithm 1, in both versions: SSVD0, using Algorithm 2, and SSVD, using Algorithm 3. For the J-orthogonal algorithm we only show the quantity that governs its error: $\Xi_\Lambda^{(S)}$. When comparing the results of routines SSVD0 and SSVD with the corresponding relative gaps of singular values (rows 1 and 3), it can be seen that both methods behave as expected. When comparing the errors in the bases computed using the routine SSVD0 with the relative gap between eigenvalues, the results can go rather poorly (see row 2).[9] When using SSVD these results improve significantly (compare rows 4 and 2), showing that the method computes the bases for these test matrices with errors depending on the relative gap between eigenvalues, as the J-orthogonal algorithm does. It can be observed that the control quantities increase mildly with n for all the algorithms. Since this effect is not observed in the accuracy of the eigenvalues, this lead us to question if it is a real effect of the eigenvector bounds or is simply reflecting the fact that the quantities Ξ are computed from n-dimensional vectors.

Table 6.4 shows the quantities $\xi_\sigma^{(S)}$ and $\xi_\lambda^{(S)}$ defined in (71). These are the quantities referring to the errors eigenvector by eigenvector. It can be seen that the accuracy of the eigenvectors is not spoiled by the clustering processes implicit in Algorithms

[9] However, as can be deduced from the mean value of $\Xi_\Lambda^{(\text{SSVD0})}$, matrices for which SSVD0 computes eigenvectors with a large error with respect to the relative gap between eigenvalues are quite infrequent.

TABLE 6.3
Experiment 1. *Statistical data for accuracy in bases of invariant subspaces.*

n	50		100		250		500		1000	
Method	mean	max	mean	max	mean	max	mean	max	mean	max
$\Xi_{\Sigma}^{(\texttt{SSVDO})}$	.032	.46	.051	1.2	.084	2.5	.12	4.5	.17	4.4
$\Xi_{\Lambda}^{(\texttt{SSVDO})}$	.37	320	1.1	3300	2.4	500	6.5	1700	5.6	150
$\Xi_{\Sigma}^{(\texttt{SSVD})}$	.034	.50	.056	1.2	.095	2.5	.13	4.5	.18	4.4
$\Xi_{\Lambda}^{(\texttt{SSVD})}$	.041	.65	.075	4.6	.15	3.2	.23	6.0	.37	7.3
$\Xi_{\Lambda}^{(\texttt{J-O})}$	.044	.64	.076	1.5	.15	2.6	.21	5.7	.32	7.3

TABLE 6.4
Experiment 1. *Statistical data for accuracy in eigenvectors:* $\xi_{\sigma}^{(\mathrm{S})}$ *and* $\xi_{\lambda}^{(\mathrm{S})}$.

n	50		100		250		500		1000	
Method	mean	max	mean	max	mean	max	mean	max	mean	max
$\xi_{\sigma}^{(\texttt{SSVDO})}$	.033	.74	.057	1.3	.092	2.5	.13	4.5	.19	4.4
$\xi_{\lambda}^{(\texttt{SSVDO})}$	.37	320	1.1	3300	2.4	500	6.5	1700	5.6	150
$\xi_{\sigma}^{(\texttt{SSVD})}$	.035	.90	.063	1.6	.10	2.5	.14	4.5	.20	4.4
$\xi_{\lambda}^{(\texttt{SSVD})}$	.045	.90	.089	4.6	.17	3.2	.26	6.0	.42	7.3
$\xi_{\lambda}^{(\texttt{J-O})}$	.044	.64	.076	1.5	.15	2.6	.21	5.7	.32	7.3

`SSVD` and `SSVDO`. Comments similar to those made with respect to Table 6.3 apply here.

To conclude, we show other quantities of numerical interest. The minimum singular value and eigenvalue relative gaps for clusters selected in Algorithm 2 have exceeded, respectively, 10^{-5} and 10^{-4}, and after the clustering process in Algorithm 3 both relative gaps, for eigenvalues and singular values, have been bigger than 10^{-4}. The minimum relative gap for individual eigenvalues has been greater than 10^{-5}, and for singular values greater than 10^{-8}. The maximum values of $\kappa(R')$ have been 190 for $n = 50$, 270 for $n = 100$, 600 for $n = 250$, 1300 for $n = 500$, and 2200 for $n = 1000$, showing that it increases roughly as some constant times n.

EXPERIMENT 2. We have generated $n \times n$ graded matrices $A = DBD$ by multiplying random well-conditioned matrices, B, and random ill-conditioned diagonal matrices, D, to test the accuracy of the complete Algorithm 1 including the factorization in step 1. Not always can an accurate RRD fulfilling (10) be computed for graded matrices [6, section 4]: the accuracy that can be guaranteed at best (and is frequently achieved in practice) is $O(\epsilon_s \kappa(B))$. Thus, high relative accuracy is expected when computing eigenvalues and eigenvectors for the matrices generated in this experiment. As mentioned in section 6.1, the initial RRD in Algorithm 1 has been done in two ways: using either a modification of the symmetric indefinite BP decomposition or a nonsymmetric LU factorization with complete pivoting. We have obtained similar results for both decompositions. Parameters have been chosen as follows: $\kappa(B) = 10^{[0:1:3]}$, $\kappa(D) = 10^{[2:2:10]}$, $MODE_B = 3, 4, 5$, $MODE_D = \pm 3, \pm 4, 5$. For each set of parameters we have run 50 matrices for $n = 50, 100$ (total 15000 matrices for each n), 5 for $n = 250, 500$ (total 1500 matrices for each n), 1 for $n = 1000$, and only for 5 combinations of the $MODE$s (total 100 matrices). As announced, Jacobi and QR also have been applied on these test matrices.

The same quantities as in Experiment 1 are shown in Table 6.5 for eigenvalues and in Table 6.6 for individual eigenvectors. The results for bases of invariant subspaces

TABLE 6.5
Experiment 2. *Statistical data for accuracy in eigenvalues:* $\vartheta^{(\mathrm{S})}$.

n	50		100		250	
Method	mean	max	mean	max	mean	max
$\vartheta^{(\texttt{SSVD})}$	1.8	2600	.82	1100	.21	52
$\vartheta^{(\texttt{J-O})}$	1.5	1100	.80	1200	.21	64
$\vartheta^{(\texttt{JAC})}$	$3 \cdot 10^{15}$	$3 \cdot 10^{19}$	$1 \cdot 10^{14}$	$3 \cdot 10^{17}$	$1 \cdot 10^{13}$	$7 \cdot 10^{15}$
$\vartheta^{(\texttt{QR})}$	$2 \cdot 10^{13}$	$2 \cdot 10^{17}$	$7 \cdot 10^{11}$	$5 \cdot 10^{15}$	$5 \cdot 10^{10}$	$4 \cdot 10^{13}$
$\vartheta^{(\texttt{SVD})}$	1.8	2600	.82	1100	.21	52

n	500		1000	
Method	mean	max	mean	max
$\vartheta^{(\texttt{SSVD})}$	.22	140	.014	.24
$\vartheta^{(\texttt{J-O})}$	.31	320	.019	.33
$\vartheta^{(\texttt{JAC})}$	$7 \cdot 10^{12}$	$5 \cdot 10^{15}$	$2 \cdot 10^{11}$	$8 \cdot 10^{12}$
$\vartheta^{(\texttt{QR})}$	$2 \cdot 10^{10}$	$1 \cdot 10^{13}$	$2 \cdot 10^{3}$	$4 \cdot 10^{4}$
$\vartheta^{(\texttt{SVD})}$	.22	140	.014	.24

TABLE 6.6
Experiment 2. *Statistical data for accuracy in eigenvectors:* $\xi_\sigma^{(\mathrm{S})}$ *and* $\xi_\lambda^{(\mathrm{S})}$.

n	50		100		250		500		1000	
Method	mean	max	mean	max	mean	max	mean	max	mean	max
$\xi_\sigma^{(\texttt{SSVDO})}$	.47	11	.28	4.6	.17	1.1	.064	.55	.023	.16
$\xi_\lambda^{(\texttt{SSVDO})}$	3.6	3300	2.8	1900	1.2	1600	.30	14	.067	.51
$\xi_\sigma^{(\texttt{SSVD})}$	.47	11	.31	5.2	.20	1.1	.076	1.3	.024	.16
$\xi_\lambda^{(\texttt{SSVD})}$	.56	12	.34	5.8	.25	2.4	.091	1.3	.030	.16
$\xi_\lambda^{(\texttt{J-O})}$	.60	21	.37	4.3	.17	1.2	.090	.67	.039	.20

are almost the same as those in Table 6.6 and, therefore, are not shown. In these tables we show only the data corresponding to symmetric RRDs obtained by the BP method. The corresponding data for these tables using the unsymmetric RRD based on GECP are so similar that they are omitted. Nevertheless for other quantities (see Tables 6.7 and 6.8) we show the results for both decompositions (GECP is abbreviated as CP in the tables).

Notice that the maximum values in Table 6.5 are greater than in Experiment 1, for both Algorithm 1 and the J-orthogonal algorithm. This is due to the error in the initial factorization step, which is roughly bounded by $O(\epsilon_s \kappa(B))$. In any case, they behave much better than the classical methods, Jacobi and QR. An interesting remark is that the quantities $\vartheta^{(\mathrm{S})}$ decrease in Table 6.5 as n increases. This is because in this experiment (see Table 6.7) the condition number κ increases with the dimension n faster than the relative errors $e_\lambda^{(S)}$ in the eigenvalues. The control quantities for eigenvectors in Table 6.6 also decrease with n for the same reason. However, the maximum values of the control quantities for eigenvalues (Table 6.5) are much bigger than those of eigenvectors (Table 6.6). This is not explained by the error bounds.

As in Experiment 1, for a good number of the generated matrices (310 matrices out of 15000 for $n = 50$; 4821 matrices out of 15000 for $n = 100$; 1019 matrices out of 1500 for $n = 250$; 1454 matrices out of 1500 for $n = 500$; 100 matrices out of 100 for $n = 1000$), there are clusters of singular values of dimension greater than 1, according to criterion (65), with a maximal dimension of 5. The average number of clusters has ranged from almost no clusters for $n = 50$ to approximately 60 clusters

TABLE 6.7
Experiment 2. *Table for* $\kappa(R')$ *and* $M_\kappa = \max\{\kappa(X), \kappa(Y)\}$.

n	50		100		250		500		1000	
Method	mean	max	mean	max	mean	max	mean	max	mean	max
$\kappa(R')$(BP)	11	39	23	84	67	220	150	430	330	960
$\kappa(R')$(CP)	11	37	24	80	71	201	160	450	360	860
$\kappa(X)$ (BP)	100	500	300	1300	1400	5000	4300	16000	14000	40000
M_κ (CP)	78	320	230	1000	1000	3200	2900	7900	5000	20000

TABLE 6.8
Experiment 2. *Statistical data for the number of sweeps.*

n	50		100		250		500		1000	
Method	mean	max	mean	max	mean	max	mean	max	mean	max
$Sweeps^{\texttt{(SSVD)BP}}$	5.0	7	5.6	8	6.4	9	7.3	9	8.1	9
$Sweeps^{\texttt{(SSVD)CP}}$	5.0	7	5.5	8	6.4	9	7.2	9	8.0	9
$Sweeps^{\texttt{(J-O)}}$	6.3	8	7.1	10	8.5	11	9.6	12	11.0	13

for $n = 1000$ with a typical dimension of 2. This shows again that criterion (65) determines perfectly in practice the signs of the eigenvalues, even when clusters are present. After applying Algorithm 3.1 all the considered matrices have clusters. The average number of clusters has been in this case around $0.3n$ for all n.

In addition, we show other quantities of numerical interest. The minimum singular value and eigenvalue relative gaps for clusters selected in Algorithm 2 are, respectively, 10^{-5} and $3.3 \cdot 10^{-4}$; and after the clustering process in Algorithm 3 both relative gaps, for eigenvalues and singular values, have reached the minimum $3.3 \cdot 10^{-4}$. The minimum relative gap for individual eigenvalues has been $4.1 \cdot 10^{-5}$, and for singular values greater than $9.1 \cdot 10^{-8}$. With respect to the condition numbers $\kappa(X)$, $\max\{\kappa(X), \kappa(Y)\}$ and $\kappa(R')$, they are shown in Table 6.7. The maximum values of $\epsilon\kappa(X)\kappa(R')$ are $8 \cdot 10^{-4}$ for $n = 50$, $4 \cdot 10^{-3}$ for $n = 100$, $5 \cdot 10^{-2}$ for $n = 250$, $3 \cdot 10^{-1}$ for $n = 500$, and 1.8 for $n = 1000$, showing that it increases roughly as some constant times n.

Table 6.8 shows that the J-orthogonal algorithm uses again more sweeps than Algorithm 1: on average, from one more for $n = 50$ to three more for $n = 1000$. This is reflected in the run-time used by the different routines. Taking as a reference the time employed by the QR routine (`SSYEV` of LAPACK), we have the following average results for our experiments: For $n = 100$, Algorithm `SSVD` (with symmetric RRD factorization) employs 200% more time than QR, the J-orthogonal algorithm employs 250% more time, and the Jacobi algorithm `SJAC` employs 190% more time; for $n = 500$, Algorithm `SSVD` (with symmetric RRD factorization) employs 380% more time, the J-orthogonal algorithm employs 350% more time, and the Jacobi algorithm `SJAC` employs 340% more time. These numbers can be explained as coming from two opposite effects: `SSVD` uses less Jacobi sweeps, but the number of clusters increases with the size of the matrix.

EXPERIMENT 3. We have also generated full matrices in another form to compare the accuracy of Algorithms 1 and J-orthogonal. We have used the matrix generator developed in [22], which is specifically designed to test the performance of the J-orthogonal algorithm on matrices for which the error bounds of this algorithm are controlled (see [22] for details).

The set of parameters has been chosen as follows: $n = 100$; `ASCAL` $= [1 : 1 : 3]$;

TABLE 6.9
Experiment 3. *Statistical data.*

	ϑ		ξ_σ		ξ_λ		*Sweeps*	
Method	mean	max	mean	max	mean	max	mean	max
`SSVD`	.27	2.2	2.1	14	2.9	21	4.6	6
`J-O`	.47	2.8	–	–	3.1	20	5.5	8

`HSCAL`$= [5 : 2 : 25]$.[10] For each set of parameters we have run 50 matrices, in total 1650 matrices.

The results confirm that Algorithm `SSVD` performs very well also for matrices of this type. The results for eigenvalues, eigenvectors, and number of sweeps are summarized in Table 6.9. As in the other experiments, the results for individual eigenvectors, $\xi_{\sigma,\lambda}^{(\mathtt{S})}$, are similar to those for bases. For this set of matrices, no clusters of singular values with dimension greater than 1 were found in the sense of criterion (65).

EXPERIMENT 4. The results for testing the accuracy of computed eigenvectors in previous experiments seem to show that the errors for the `SSVD` and J-orthogonal algorithms are comparable (see rows 4 and 5 of Tables 6.4, 6.6 and columns 6–7 of Table 6.9 in Experiment 3), both depending on the relative gap between eigenvalues. However, it should not be forgotten that the error bound for eigenvectors in the `SSVD` algorithm is given by the expressions (4) and (5) (or, more precisely, Theorem 5.12) and not (11). It is not difficult to think of situations in which Algorithm `SSVD` can calculate single eigenvectors much worse than the J-orthogonal algorithm. Take for example the following 3×3 very well conditioned matrix generated in single precision:

$$A = \begin{bmatrix} .1804019 & .9148742 & -.3611555 \\ .9148742 & -.2908984 & -.2799287 \\ -.3611555 & -.2799287 & -.8894936 \end{bmatrix}$$

with eigenvalues $\lambda_1 = 0.9999904633563307$, $\lambda_2 = -0.9999802814301686$, and $\lambda_3 = -1.000000302456291$ in double precision. The corresponding computed eigenvectors in single precision have the following errors for the `SSVD` algorithm:

$$[\|q_i - q_i^{(\mathrm{SSVD})}\|_2]_{i=1,2,3} = [3.12,\ 5.25,\ 4.23] \times 10^{-3}$$

and

$$[\|q_i - q_i^{(\mathrm{J-O})}\|_2]_{i=1,2,3} = [3.79 \times 10^{-5},\ 1.43,\ 1.43] \times 10^{-3}$$

for the J-orthogonal algorithm. Notice that the J-orthogonal algorithm computes the eigenvector corresponding to the positive eigenvalue λ_1 with full machine precision, while with the `SSVD` algorithm five significant decimal digits are lost. The reason for this is easily understood, because the eigenvalue relative gap for λ_1 is 1, while the corresponding singular value relative gap is near 10^{-5} (in this case relative or absolute gaps are equivalent). This cannot be improved by the clustering process done in Algorithm 3.1, because any of the two possible clusters of singular values containing one positive and one negative eigenvalue has a close singular value at a distance of order 10^{-5}, and the minimum of the eigenvalue relative gaps is also of order 10^{-5}.

[10] The routine `GENSYM` generates a nonsingular symmetric matrix H of order n, with $\kappa(H) \approx 10^{\mathtt{HSCAL}}$ and the measure $C(A, \widehat{A}) \approx 10^{\mathtt{ASCAL}}$ (see [22] for details).

However, notice that the SSVD algorithm is able to compute all the eigenvectors with three correct decimal digits and that $\max_i e_{q_i}^{(\mathtt{SSVD})} / \max_i e_{q_i}^{(\mathtt{J\text{-}O})} = 3.7$, of order 1 as predicted by the bound (5); i.e., the J-orthogonal algorithm also computes some eigenvectors with three correct significant digits.

Finally, notice that if all the eigenvalues of the matrix A are considered inside the same cluster, the SSVD algorithm computes the eigenvector corresponding to λ_1 with full machine precision, according to the bound (51). However, the eigenvectors corresponding to the negative eigenvalues are computed with errors of order 1, although according to (51) they form a very accurate orthonormal basis of the invariant subspace associated with the negative eigenvalues.

EXPERIMENT 5. Our last experiment is designed to show how the SSVD algorithm, like the J-orthogonal one, is able to compute accurate bases of invariant subspaces, even when the gaps between eigenvalues are very small.

We generate a 10×10 matrix $A = QDQ^T$ by multiplying, in single precision, a single precision random orthogonal matrix Q by the diagonal matrix $D = \mathrm{diag}[-1, 1, 1, 1, 1, 0.1, 0.1, 0.1, 0.1, 0.1]$. Due to roundoff errors, the absolute values of all the eigenvalues of A become different. But two clusters of singular values are found according to criterion (65), one around 1, of dimension 5, and another around 0.1, of the same dimension. Since one of the clusters is unsigned, Algorithm 3.1 does not change these clusters. The absolute gaps between the singular values inside each cluster exceed 10^{-7}. Thus the double precision routine DSYEVJ computes the eigenvectors with at least eight correct decimal digits. The SSVD and J-orthogonal algorithms, in single precision, compute all the eigenvectors with errors of $O(1)$, except the eigenvector corresponding to the negative eigenvalue which is computed, in both cases, with an error near 10^{-7}. This error is predicted by bound (51) for the SSVD algorithm (see also the remarks after the proof of Theorem 4.7). The errors in the invariant subspaces can be estimated using $E_{\Lambda_i}^{(\mathrm{S})}$ in (68). These, for SSVD and J-orthogonal algorithms, are of order 10^{-7} for the following invariant subspaces: the subspace corresponding to the four positive eigenvalues close to 1; the subspace corresponding to the five positive eigenvalues close to 0.1; and the subspace corresponding to the negative eigenvalue. Moreover, the same errors appear if we consider the invariant subspace corresponding to all the eigenvalues of absolute value around 1 (including the negative one). This shows in practice that, as studied in the error analysis leading to Theorem 4.7, once a cluster of singular values is chosen, we obtain two bases, one for the invariant subspace corresponding to the positive eigenvalues in the cluster and another for the negative ones, with an error of the same order as the one appearing in the basis of the singular subspace corresponding to the whole cluster of singular values.

7. Conclusions and future work. In this paper we have presented formal error analysis and numerical experiments of a new algorithm which computes eigenvalues and eigenvectors with high relative accuracy for the largest class of symmetric matrices known so far—in particular for all symmetric matrices belonging to the classes of general matrices studied in [6]. This high relative accuracy is achieved for a given symmetric matrix A whenever an accurate rank-revealing decomposition (RRD) of A can be computed.

The new algorithm is based on computing, in a first stage, a singular value decomposition (SVD) of the symmetric matrix A. This is the reason for its wide applicability, because in this stage the symmetry of A is not used. Thus, we can compute nonsymmetric RRDs of A and apply the theory developed in [6].

It is not known if accurate symmetric RRDs can be computed for all symmetric

matrices in any of the classes described in [6]. The J-orthogonal algorithm [26, 22] computes eigenvalues and eigenvectors with high relative accuracy *only* if symmetric RRDs that are accurate enough are available. The authors are presently studying this interesting question.

Appendix. Proof of Theorem 5.7. We begin with some previous elementary results that will be frequently used.

Let a and a' be any two real numbers. Then

$$\frac{a-a'}{a'} = \frac{\frac{a-a'}{a}}{1-\frac{a-a'}{a}} \quad \text{and} \quad \frac{a}{a'} = \frac{1}{1-\frac{a-a'}{a}}. \tag{72}$$

The following lemma bounds the relative distance between the maximum and the minimum elements in a cluster of tolerance C_l.

LEMMA A.1. *Let $\Sigma_1 = \{\sigma_{i+1}, \sigma_{i+2}, \ldots, \sigma_{i+d_1}\}$ be a cluster of tolerance C_l with d_1 elements. Then*

$$\frac{\sigma_{i+1} - \sigma_{i+d_1}}{\sigma_{i+1}} \leq (d_1 - 1)\, C_l.$$

Proof. Notice that

$$\frac{\sigma_{i+1} - \sigma_{i+d_1}}{\sigma_{i+1}} = \frac{\sigma_{i+1} - \sigma_{i+2}}{\sigma_{i+1}} + \frac{\sigma_{i+2} - \sigma_{i+3}}{\sigma_{i+1}} + \cdots + \frac{\sigma_{i+d_1-1} - \sigma_{i+d_1}}{\sigma_{i+1}}.$$

Thus

$$\frac{\sigma_{i+1} - \sigma_{i+d_1}}{\sigma_{i+1}} \leq \frac{\sigma_{i+1} - \sigma_{i+2}}{\sigma_{i+1}} + \frac{\sigma_{i+2} - \sigma_{i+3}}{\sigma_{i+2}} + \cdots + \frac{\sigma_{i+d_1-1} - \sigma_{i+d_1}}{\sigma_{i+d_1-1}} \leq (d_1 - 1)\, C_l.$$

∎

Proof of Theorem 5.7. Let

$$\Sigma_1 = \{\sigma_{i+1}, \sigma_{i+2}, \ldots, \sigma_{i+d_1}\}, \quad \Sigma_2 = \{\sigma_{i+d_1+1}, \sigma_{i+d_1+2}, \ldots, \sigma_{i+d_1+d_2}\} \tag{73}$$

be the two clusters of singular values appearing in the statement of the theorem. Although in this setting the elements of Σ_1 are greater than the elements of Σ_2, the opposite case can be proved with the notation in (73) by interchanging the roles of Σ_1 and Σ_2.

Lemma 5.3 implies

$$rg(\Sigma_1 \cup \Sigma_2) = \min\left\{\frac{\sigma_i - \sigma_{i+1}}{\sigma_{i+1}}, \frac{\sigma_{i+d_1+d_2} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1+d_2}}\right\}, \tag{74}$$

and

$$\begin{aligned} &\min\{rg(\Sigma_1), rg(\Sigma_2)\} \\ &\quad = \min\left\{\frac{\sigma_i - \sigma_{i+1}}{\sigma_{i+1}}, \frac{\sigma_{i+d_1} - \sigma_{i+d_1+1}}{\sigma_{i+d_1}}, \frac{\sigma_{i+d_1} - \sigma_{i+d_1+1}}{\sigma_{i+d_1+1}}, \frac{\sigma_{i+d_1+d_2} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1+d_2}}\right\}, \end{aligned} \tag{75}$$

where if some of the subindices do not belong to $\{1, \ldots, n\}$, the corresponding fraction does not appear. Therefore $rg(\Sigma_1 \cup \Sigma_2) \geq \min\{rg(\Sigma_1), rg(\Sigma_2)\}$, and the assumption (58) appearing in Theorem 5.7 leads to the following results:

1.

$$\min\{rg(\Sigma_1), rg(\Sigma_2)\} = \frac{\sigma_{i+d_1} - \sigma_{i+d_1+1}}{\sigma_{i+d_1}}. \tag{76}$$

2.

$$rg(\Sigma_1) = \frac{\sigma_{i+d_1} - \sigma_{i+d_1+1}}{\sigma_{i+d_1}}. \tag{77}$$

Thus in the setting (73), condition (58) implies that Σ_2 is the relative closest cluster to Σ_1 and it is not necessary to impose this condition explicitly. This has been done in the statement of Theorem 5.7 for the sake of clarity. Recall that one of the hypotheses of Theorem 5.7 is

$$rg(\Sigma_1) < t < 1. \tag{78}$$

The previous setting also allows us to prove Theorem 5.7 in the case in which the elements of Σ_1 are smaller than the elements of Σ_2 just by interchanging the roles of Σ_1 and Σ_2 in the statement of the theorem. Notice that condition $\mathrm{rg}\{\Sigma_1 \cup \Sigma_2\} > \min\{\mathrm{rg}\{\Sigma_1\}, \mathrm{rg}\{\Sigma_2\}\}$ remains unchanged, and therefore its consequences (76), (77) still hold. This, together with $rg(\Sigma_2) < t < 1$, leads to $rg(\Sigma_1) < t$, i.e., condition (78). Therefore, in the rest of the proof we will focus on the situation in (73) with assumptions (58) (and its consequences (76)–(77)) and (78).

Suppose that $(i + d_1 + d_2 + 1) \in \{1, \ldots, n\}$. If $\lambda_{\Pi(i+d_1+d_2+1)}$ is either zero or has the same sign as the elements of Λ_2, then $rg(\Lambda_2) \leq (\sigma_{i+d_1+d_2} - \sigma_{i+d_1+d_2+1})/\sigma_{i+d_1+d_2}$. Otherwise $\lambda_{\Pi(i+d_1+d_2+1)}$ has the same sign as the elements of Λ_1, and then $rg(\Lambda_1) \leq (\sigma_{i+d_1} - \sigma_{i+d_1+d_2+1})/\sigma_{i+d_1}$. In any case

$$\begin{aligned} \min\{rg(\Lambda_1), rg(\Lambda_2)\} &\leq \max\left\{ \frac{\sigma_{i+d_1} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1}}, \frac{\sigma_{i+d_1+d_2} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1+d_2}} \right\} \\ &= \frac{\sigma_{i+d_1} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1}}. \end{aligned} \tag{79}$$

Suppose now that i belongs to the set $\{1, \ldots, n\}$. If $\lambda_{\Pi(i)}$ has the same sign as the elements of Λ_1, then $rg(\Lambda_1) \leq (\sigma_i - \sigma_{i+1})/\sigma_{i+1}$. Otherwise $\lambda_{\Pi(i)}$ has the same sign as the elements of Λ_2, and then $rg(\Lambda_2) \leq (\sigma_i - \sigma_{i+d_1+1})/\sigma_{i+d_1+1}$. In any case

$$\min\{rg(\Lambda_1), rg(\Lambda_2)\} \leq \max\left\{ \frac{\sigma_i - \sigma_{i+1}}{\sigma_{i+1}}, \frac{\sigma_i - \sigma_{i+d_1+1}}{\sigma_{i+d_1+1}} \right\} = \frac{\sigma_i - \sigma_{i+d_1+1}}{\sigma_{i+d_1+1}}. \tag{80}$$

Once (79) and (80) have been established, it only remains to prove

$$\frac{\sigma_{i+d_1} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1}} \leq R \, \frac{\sigma_{i+d_1+d_2} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1+d_2}} \tag{81}$$

and

$$\frac{\sigma_i - \sigma_{i+d_1+1}}{\sigma_{i+d_1+1}} \leq R \, \frac{\sigma_i - \sigma_{i+1}}{\sigma_{i+1}}, \tag{82}$$

where

$$R = \frac{1}{1-t}\left(1 + \frac{1}{1-(d-1)\,C_l} + \frac{1}{1-(d-1)\,C_l}\,\frac{(d-1)\,C_l}{rg(\Sigma_1 \cup \Sigma_2)}\right).$$

If these two inequalities hold, then (79) and (80) imply that $\min\{rg(\Lambda_1), rg(\Lambda_2)\}$ is bounded simultaneously by the right-hand side of (81) and the right-hand side of (82). Thus using (74), Theorem 5.7 is finally proved.

Proof of (81). Notice that

(83)
$$\frac{\sigma_{i+d_1} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1}} = \frac{\sigma_{i+d_1} - \sigma_{i+d_1+1}}{\sigma_{i+d_1}} + \frac{\sigma_{i+d_1+1} - \sigma_{i+d_1+d_2}}{\sigma_{i+d_1}} + \frac{\sigma_{i+d_1+d_2} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1}}.$$

The first term of the right-hand side in the previous equation is less than $(\sigma_{i+d_1+d_2} - \sigma_{i+d_1+d_2+1})/\sigma_{i+d_1+d_2}$, due to (76) and (75). The third term is trivially bounded by the same quantity, since $\sigma_{i+d_1} > \sigma_{i+d_1+d_2}$. For the second term,

$$\frac{\sigma_{i+d_1+1} - \sigma_{i+d_1+d_2}}{\sigma_{i+d_1}} < \frac{\sigma_{i+d_1+1} - \sigma_{i+d_1+d_2}}{\sigma_{i+d_1+1}} \leq (d_2 - 1)C_l,$$

where the last inequality is just Lemma A.1 applied to Σ_2. Plugging these bounds into (83) and using $rg(\Sigma_1 \cup \Sigma_2) \leq (\sigma_{i+d_1+d_2} - \sigma_{i+d_1+d_2+1})/\sigma_{i+d_1+d_2}$, we obtain

$$\frac{\sigma_{i+d_1} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1}} \leq \left(2 + \frac{(d_2 - 1)C_l}{rg(\Sigma_1 \cup \Sigma_2)}\right) \frac{\sigma_{i+d_1+d_2} - \sigma_{i+d_1+d_2+1}}{\sigma_{i+d_1+d_2}}.$$

The first factor of the right-hand side is bounded by R and (81) follows. ∎

Proof of (82). Notice that

$$\frac{\sigma_i - \sigma_{i+d_1+1}}{\sigma_{i+d_1+1}} = \frac{\sigma_i - \sigma_{i+1}}{\sigma_{i+d_1+1}} + \frac{\sigma_{i+1} - \sigma_{i+d_1}}{\sigma_{i+d_1+1}} + \frac{\sigma_{i+d_1} - \sigma_{i+d_1+1}}{\sigma_{i+d_1+1}}. \tag{84}$$

Now we will bound the three terms in the right-hand side of (84). We begin with the last one: using the first equality in (72), (77), (78), and (76), we get

$$\frac{\sigma_{i+d_1} - \sigma_{i+d_1+1}}{\sigma_{i+d_1+1}} < \frac{1}{1-t} \frac{\sigma_{i+d_1} - \sigma_{i+d_1+1}}{\sigma_{i+d_1}} < \frac{1}{1-t} \frac{\sigma_i - \sigma_{i+1}}{\sigma_{i+1}}. \tag{85}$$

For the second term, the first equality in (72) and Lemma A.1 yield

$$\frac{\sigma_{i+1} - \sigma_{i+d_1}}{\sigma_{i+d_1+1}} = \frac{\sigma_{i+d_1}}{\sigma_{i+d_1+1}} \frac{\frac{\sigma_{i+1} - \sigma_{i+d_1}}{\sigma_{i+1}}}{1 - \frac{\sigma_{i+1} - \sigma_{i+d_1}}{\sigma_{i+1}}} \leq \frac{\sigma_{i+d_1}}{\sigma_{i+d_1+1}} \frac{(d_1 - 1)C_l}{1 - (d_1 - 1)C_l}.$$

The factor $\sigma_{i+d_1}/\sigma_{i+d_1+1}$ can be bounded by $1/(1-t)$, using the second equality in (72), (77), and (78). Therefore, the following bound for the second term of the right-hand side of (84) is obtained:

$$\frac{\sigma_{i+1} - \sigma_{i+d_1}}{\sigma_{i+d_1+1}} \leq \frac{1}{1-t} \frac{(d_1 - 1)C_l}{1 - (d_1 - 1)C_l}. \tag{86}$$

Finally, the first term verifies

$$\frac{\sigma_i - \sigma_{i+1}}{\sigma_{i+d_1+1}} = \frac{\sigma_{i+d_1}}{\sigma_{i+d_1+1}} \frac{\sigma_{i+1}}{\sigma_{i+d_1}} \frac{\sigma_i - \sigma_{i+1}}{\sigma_{i+1}}.$$

The factor $\sigma_{i+d_1}/\sigma_{i+d_1+1}$ already has been bounded by $1/(1-t)$, while the factor $\sigma_{i+1}/\sigma_{i+d_1}$ is bounded by $1/(1-(d_1-1)C_l)$ by the second equality in (72) and Lemma A.1. Thus

$$\frac{\sigma_i-\sigma_{i+1}}{\sigma_{i+d_1+1}} \leq \frac{1}{1-t}\,\frac{1}{1-(d_1-1)C_l}\,\frac{\sigma_i-\sigma_{i+1}}{\sigma_{i+1}}. \tag{87}$$

Replacing (87), (86), and (85) in (84), and taking into account that $rg(\Sigma_1 \cup \Sigma_2) \leq (\sigma_i-\sigma_{i+1})/\sigma_{i+1}$,

$$\frac{\sigma_i-\sigma_{i+d_1+1}}{\sigma_{i+d_1+1}} \leq \frac{1}{1-t}\left(1+\frac{1}{1-(d_1-1)\,C_l}+\frac{1}{1-(d_1-1)\,C_l}\,\frac{(d_1-1)\,C_l}{rg(\Sigma_1\cup\Sigma_2)}\right)\frac{\sigma_i-\sigma_{i+1}}{\sigma_{i+1}}$$

is obtained. Now inequality (82) is easily proved. □

Acknowledgments. The authors thank Professor Zlatko Drmač, who provided the source code for the one-sided Jacobi SVD routine employed in the experiments. As can be seen in the numerical tests in section 6, the performance of his code is excellent. The authors thank also Professor J. W. Demmel for providing the source code of the routines used in [5].

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.

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

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

[4] C. Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation.* III, SIAM J. Numer. Anal., 7 (1970), pp. 1–46.

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

[6] J. 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.

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

[8] F. M. Dopico, *A note on sin Θ theorems for singular subspace variations*, BIT, 40 (2000), pp. 395–403.

[9] F. M. Dopico and J. Moro, *Perturbation theory for simultaneous bases of singular subspaces*, BIT, 42 (2002), pp. 84–109.

[10] F. M. Dopico, J. M. Molera, and J. Moro, *An Orthogonal High Relative Accuracy Algorithm for the Symmetric Eigenproblem*, Tech. report, available online at http://www.uc3m.es/uc3m/dpto/MATEM/molera/indice.html.

[11] F. M. Dopico, J. M. Molera, and J. Moro, *A note on multiplicative backward errors of accurate SVD algorithms*, submitted.

[12] Z. Drmač, *Implementation of Jacobi rotations for accurate singular value computation in floating point arithmetic*, SIAM J. Sci. Comput., 18 (1997), pp. 1200–1222.

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

[14] Z. Drmač, *A posteriori computation of the singular vectors in a preconditioned Jacobi SVD algorithm*, IMA J. Numer. Anal., 19 (1999), pp. 191–213.

[15] Z. Drmač and K. Veselić, *Approximate eigenvectors as preconditioner*, Linear Algebra Appl., 309 (2000), pp. 191–215.

[16] S. C. Eisenstat and I. C. F. Ipsen, *Relative perturbation techniques for singular value problems*, SIAM J. Numer. Anal., 32 (1995), pp. 1972–1988.

[17] G. H. Golub and C. F. Van Loan, *Matrix Computations,* 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.
[18] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.
[19] R.-C. Li, *Relative perturbation theory:* I. *Eigenvalue and singular value variations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 956–982.
[20] R.-C. Li, *Relative perturbation theory:* II. *Eigenspace and singular subspace variations*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 471–492.
[21] R. Mathias, *Accurate eigensystem computations by Jacobi methods*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 977–1003.
[22] I. Slapničar, *Accurate Symmetric Eigenreduction by a Jacobi Method*, Ph.D. thesis, Fachbereich Mathematik Fernuniversität, Gesamthochschule Hagen, Germany, 1992.
[23] I. Slapničar, *Componentwise analysis of direct factorization of real symmetric and Hermitian matrices*, Linear Algebra Appl., 272 (1998), pp. 227–275.
[24] A. van der Sluis, *Condition numbers and equilibration of matrices*, Numer. Math., 14 (1969), pp. 14–23.
[25] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, New York, 1990.
[26] K. Veselić, *A Jacobi eigenreduction algorithm for definite matrix pairs*, Numer. Math., 64 (1993), pp. 241–269.
[27] K. Veselić and I. Slapničar, *Floating-point perturbations of Hermitian matrices*, Linear Algebra Appl., 195 (1993), pp. 81–116.