Electronic Transactions on Numerical Analysis.
Volume 39, pp. 1-21, 2012.
Copyright © 2012, Kent State University.
ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

THE MR^3-GK ALGORITHM FOR THE BIDIAGONAL SVD[*]

PAUL R. WILLEMS[†] AND BRUNO LANG[‡]

Abstract. Determining the singular value decomposition of a bidiagonal matrix is a frequent subtask in numerical computations. We shed new light on a long-known way to utilize the algorithm of multiple relatively robust representations, MR^3, for this task by casting the singular value problem in terms of a suitable tridiagonal symmetric eigenproblem (via the Golub–Kahan matrix). Just running MR^3 "as is" on the tridiagonal problem does not work, as has been observed before (e.g., by B. Großer and B. Lang [Linear Algebra Appl., 358 (2003), pp. 45–70]). In this paper we give more detailed explanations for the problems with running MR^3 as a black box solver on the Golub–Kahan matrix. We show that, in contrast to standing opinion, MR^3 *can* be run safely on the Golub–Kahan matrix, with just a minor modification. A proof including error bounds is given for this claim.

Key words. bidiagonal matrix, singular value decomposition, MRRR algorithm, theory and implementation, Golub–Kahan matrix

AMS subject classifications. 65F30, 65F15, 65G50, 15A18

1. Introduction.

The singular value decomposition (SVD) is one of the most fundamental and powerful decompositions in numerical linear algebra. This is partly due to generality, since every complex rectangular matrix has a SVD, but also to versatility, because many problems can be cast in terms of the SVD of a certain related matrix. Applications range from pure theory to image processing.

The principal algorithm for computing the SVD of an arbitrary dense complex rectangular matrix is reduction to real bidiagonal form using unitary similarity transformations, followed by computing the SVD of the obtained bidiagonal matrix. The method to do the reduction was pioneered by Golub and Kahan [18]; later improvements include reorganization to do most of the work within BLAS3 calls [1, 2, 27].

We call the problem to compute the singular value decomposition of a bidiagonal matrix BSVD. There is a long tradition of solving singular value problems by casting them into related symmetric eigenproblems. For BSVD this leads to a variety of tridiagonal symmetric eigenproblems (TSEPs). Several methods are available for solving the TSEP, including QR iteration [15, 16], bisection and inverse iteration (BI), divide and conquer [3, 22], and, most recently, the algorithm of *multiple relatively robust representations* [6, 7, 8], in short MRRR or MR^3. The latter offers to compute k eigenpairs $(\lambda_i, \mathsf{q}_i)$, $\|\mathsf{q}_i\| = 1$, of a symmetric tridiagonal matrix $\mathsf{T} \in \mathbb{R}^{n\times n}$ in (optimal) time $\mathcal{O}(kn)$, and thus it is an order of magnitude faster than BI. In addition, MR^3 requires no communication for Gram–Schmidt reorthogonalization, which opens better possibilities for parallelization. It is therefore natural and tempting to solve the BSVD problem using the MR^3 algorithm, to benefit from its many desirable features. How to do so stably and efficiently is the focus of this paper.

The remainder of the paper is organized as follows. In Section 2 we briefly review the MR^3 algorithm for the tridiagonal symmetric eigenproblem and the requirements for its correctness. The reader will need some familiarity with the core MR^3 algorithm, as described in Algorithm 2.1 and Figure 2.1, to follow the arguments in the subsequent sections. In Section 3 we turn to the BSVD. We specify the problem to be solved formally, introduce the

[†] (willems@math.uni-wuppertal.de).

[‡] University of Wuppertal, Faculty of Mathematics and Natural Sciences, Gaußstr. 20, D-42097 Wuppertal (lang@math.uni-wuppertal.de).

[*] Received November 25, 2011. Accepted January 3, 2012. Published online March 5, 2012. Recommended by M. Hochstenbach. This work was carried out while P. Willems was with the Faculty of Mathematics and Natural Sciences at the University of Wuppertal. The research was partially funded by the Bundesministerium für Bildung und Forschung, contract number 01 IH 08 007 B, within the project *ELPA—Eigenwert-Löser für Petaflop-Anwendungen*.

associated tridiagonal problems, and set up some notational conventions. Invoking MR^3 on symmetric tridiagonal matrices of even dimension that have a zero diagonal, so-called *Golub–Kahan matrices*, will be investigated in Section 4. Finally, Section 5 contains numerical experiments to evaluate our implementation.

The idea of using the MR^3 algorithm for the BSVD by considering suitable TSEPs is not new. A previous approach [19, 20, 21, 39] "couples" the three TSEPs involving the normal equations and the Golub–Kahan matrix in a way that ensures good orthogonality of the singular vectors *and* small residuals; see also Section 3.3.1. For a long time the standing opinion was that using MR^3 (or any other TSEP solver) on the Golub–Kahan matrix alone is fundamentally flawed. In this paper we refute that notion, at least with regard to MR^3. Indeed we provide a complete proof, including error bounds, showing that just a minor modification makes using MR^3 on the Golub–Kahan matrix a valid solution strategy for BSVD. This method is much simpler to implement and analyze than the coupling-based approach; in particular, all levels in the MR^3 representation tree (Figure 2.1) can be handled in a uniform way.

Before proceeding we want to mention that an alternative and highly competitive solution strategy for the SVD was only recently discovered by Drmač and Veselić [10, 11]. Their method first reduces a general matrix A to non-singular triangular form via rank-revealing QR factorizations, and then an optimized version of Jacobi's iteration is applied to the triangular matrix, making heavy use of the structure to save on operations and memory accesses. Compared to methods involving bidiagonal reduction, this new approach can attain better accuracy for certain classes of matrices (e.g., if $\mathsf{A} = \widetilde{\mathsf{A}}\mathsf{D}$ with a diagonal "scaling" matrix D, then the achievable precision for the tiny singular values is determined by the condition number $\kappa_2(\widetilde{\mathsf{A}})$ instead of $\kappa_2(\mathsf{A})$, which may be considerably worse). Numerical experiments in [10, 11] also indicate that the new method tends to be somewhat faster than bidiagonal reduction followed by QR iteration on the bidiagonal matrix, but slightly slower than bidiagonal reduction and bidiagonal divide and conquer, in particular for larger matrices. As multi-step bidiagonalization (similarly to [2]) and replacing divide and conquer with the MR^3 algorithm may further speed up the bidiagonalization-based methods, the increased accuracy currently seems to come with a penalty in performance.

2. The MR^3 algorithm for the tridiagonal symmetric eigenproblem.

The present paper relies heavily on the MR^3 algorithm for TSEP and on its properties. A generic version of the algorithm has been presented in [35, 37], together with a proof that the eigensystems computed by MR^3 feature small residuals and sufficient orthogonality if five key requirements are fulfilled. In order to make the following exposition self-contained we briefly repeat some of the discussion on MR^3 from [37]; for details and proofs the reader is referred to that paper. Along the way we also introduce notation that will be used in the subsequent sections.

2.1. The algorithm.

The "core" of the MR^3 method is summarized in Algorithm 2.1. In each pass of the main loop, the algorithm considers a symmetric tridiagonal matrix, which is represented by some data M, and tries to compute specified eigenpairs $(\lambda_i, \mathsf{q}_i)$, $i \in I$. First, the eigenvalues of the matrix are determined to such precision that they can be classified as *singletons* (with sufficient *relative* distance to the other eigenvalues, e.g., agreement to at most three leading decimal digits if $gaptol \sim 10^{-3}$) and *clusters*. For singletons λ_i, a variant of a Rayleigh quotient iteration (RQI) and inverse iteration yields an accurate eigenpair. Clusters $\lambda_i \approx \ldots \approx \lambda_{i+s}$ cannot be handled directly. Instead, for each cluster one chooses a *shift* $\tau \approx \lambda_i$ very close to (or even inside) the cluster and considers the matrix $\mathsf{T} - \tau\mathsf{I}$. The eigenvalues $\lambda_i - \tau, \ldots, \lambda_{i+s} - \tau$ of that matrix will then feature much larger relative distances than $\lambda_i, \ldots, \lambda_{i+s}$ did, and therefore they may be singletons for $\mathsf{T} - \tau\mathsf{I}$, meaning that now eigenvectors can be computed in a reliable way. If some of these eigenvalues are

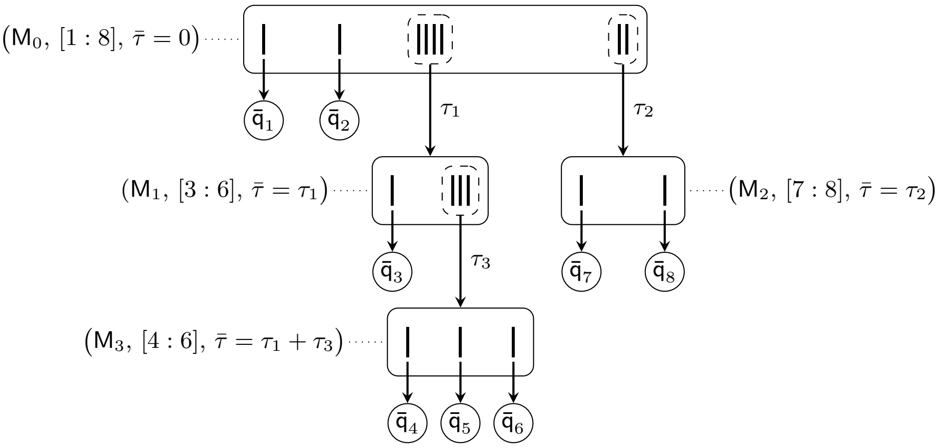

FIG. 2.1. *Example for a representation tree. The leaves corresponding to the computation of eigenvectors are not considered to be nodes. Thus the tree contains only four nodes, and the eigenpair* $(\bar\lambda_3, \bar{\mathsf{q}}_3)$ *is computed at node* $(\mathsf{M}_1, [3:6], \bar\tau)$.

still clustered, then the shifting is repeated. (To avoid special treatment, the original matrix T is also considered to be shifted with $\bar\tau = 0$.) Proceeding this way amounts to traversing a so-called *representation tree* with the original matrix T at the root, and children of a node standing for shifted matrices due to clusters; see Figure 2.1 for an example. The computation of eigenvectors corresponds to the leaves of the tree.

2.2. Representations of tridiagonal symmetric matrices. The name MR^3 comes from the fact that the transition from a node to its child, $\mathsf{M} - \tau =: \mathsf{M}^+$, must not change the invariant subspace of a cluster—and at least some of its eigenvalues—by too much (see Requirement RRR in Section 2.5). In general, this robustness cannot be achieved if the tridiagonal matrices are represented by their $2n-1$ entries because those do not necessarily determine small eigenvalues to high relative precision. Therefore other representations are used, e.g., lower (upper) bidiagonal factorizations $\mathsf{T} = \mathsf{LDL}^*$ ($\mathsf{T} = \mathsf{URU}^*$, resp.) with

$$\begin{array}{rcll}
\mathsf{D} & = & \operatorname{diag}(d_1, \dots, d_n) & \text{diagonal,} \\
\mathsf{L} & = & \operatorname{diag}(1, \dots, 1) + \operatorname{diag}_{-1}(\ell_1, \dots, \ell_{n-1}) & \text{lower bidiagonal,} \\
\mathsf{R} & = & \operatorname{diag}(r_1, \dots, r_n) & \text{diagonal, and} \\
\mathsf{U} & = & \operatorname{diag}(1, \dots, 1) + \operatorname{diag}_{+1}(u_2, \dots, u_n) & \text{upper bidiagonal.}
\end{array}$$

Note that we write * for the transpose of a matrix. The so-called *twisted* factorizations

$$\mathsf{T} \; = \; \mathsf{N}_k \mathsf{G}_k \mathsf{N}_k^*$$

with

$$\text{(2.1)} \qquad \mathsf{N}_k \; = \; \begin{bmatrix}
1 & & & & & & \\
\ell_1 & 1 & & & & & \\
 & \ddots & \ddots & & & & \\
 & & \ell_{k-1} & 1 & u_{k+1} & & \\
 & & & & \ddots & \ddots & \\
 & & & & & 1 & u_n \\
 & & & & & & 1
\end{bmatrix}, \quad \mathsf{G}_k \; = \; \begin{bmatrix}
d_1 & & & & & & \\
 & \ddots & & & & & \\
 & & d_{k-1} & & & & \\
 & & & \gamma_k & & & \\
 & & & & r_{k+1} & & \\
 & & & & & \ddots & \\
 & & & & & & r_n
\end{bmatrix}$$

```
Input:      Symmetric tridiagonal $\mathsf{T} \in \mathbb{R}^{n\times n}$, index set $I_0 \subseteq \{1,\dots,n\}$
Output:     Eigenpairs $(\bar\lambda_i, \bar{\mathsf{q}}_i), i \in I_0$
Parameter:  $gaptol$, the gap tolerance

Find a suitable representation $\mathsf{M}_0$ for $\mathsf{T}$, preferably definite, possibly by shifting $\mathsf{T}$.
$\mathcal{S} := \{(\mathsf{M}_0, I_0, \bar\tau = 0)\}$
while $\mathcal{S} \neq \emptyset$ do
    Remove one node $(\mathsf{M}, I, \bar\tau)$ from $\mathcal{S}$
    Approximate eigenvalues $[\lambda_i^{\mathrm{loc}}]$, $i \in I$, of $\mathsf{M}$ such that they can be classified into
    singletons and clusters according to $gaptol$; this gives a partition $I = I_1 \cup \cdots \cup I_m$.
    for $r = 1$ to $m$ do
        if $I_r = \{i\}$ then          // singleton
            Refine eigenvalue approximation $[\lambda_i^{\mathrm{loc}}]$ and use it to compute $\bar{\mathsf{q}}_i$.
            If necessary iterate until the residual of $\bar{\mathsf{q}}_i$ becomes small enough,
            using a Rayleigh quotient iteration (RQI).
            $\bar\lambda_i := \lambda_i^{\mathrm{loc}} + \bar\tau$
        else                           // cluster
            Refine the eigenvalue approximations at the borders of (and/or inside) the
            cluster if desired for more accurate selection of shift.
            Choose a suitable shift $\tau$ near the cluster and compute a representation
            of $\mathsf{M}^+ = \mathsf{M} - \tau$.
            Add new node $(\mathsf{M}^+, I_r, \bar\tau + \tau)$ to $\mathcal{S}$.
        endif
    endfor
endwhile
```

Algorithm 2.1: MR^3 for TSEP: Compute selected eigenpairs of a symmetric tridiagonal T.

generalize the bidiagonal factorizations. They are built by combining the upper part of an LDL^* factorization and the lower part of a URU^* factorization, together with the *twist element* $\gamma_k = d_k + r_k - \mathsf{T}(k,k)$ at *twist index* k.

Twisted factorizations are preferred because, in addition to yielding better relative sensitivity, they also allow to compute highly accurate eigenvectors [6]. qd algorithms are used for shifting the factorizations, e.g., $\mathsf{LDL}^* - \tau\mathsf{I} \ =: \ \mathsf{L}^+\mathsf{D}^+(\mathsf{L}^+)^*$, possibly converting between them as in $\mathsf{URU}^* - \tau\mathsf{I} \ =: \ \mathsf{L}^+\mathsf{D}^+(\mathsf{L}^+)^*$.

The bidiagonal and twisted factorizations can rely on different data items being stored. To give an example, the matrix $\mathsf{T} = \mathsf{LDL}^*$ with unit lower bidiagonal L and diagonal D is defined by fixing the diagonal entries $d_1,\dots,d_n$ of D and the subdiagonal entries $\ell_1,\dots,\ell_{n-1}$ of L. We might as well use the offdiagonal entries $\mathsf{T}(1,2),\dots,\mathsf{T}(n-1,n)$, together with $d_1,\dots,d_n$, to describe the tridiagonal matrix and the factorization because the ℓ_i can be recovered from the relation $\mathsf{T}(i,i+1) = \ell_i d_i$. The question of which data one should actually use to define a matrix leads to the concept of representations.

DEFINITION 2.1. *A* representation M *of a symmetric tridiagonal matrix* $\mathsf{T} \in \mathbb{R}^{n\times n}$ *is a set of* $m \le 2n-1$ *scalars, called the* primary data, *together with a mapping* $f : \mathbb{R}^m \to \mathbb{R}^{2n-1}$ *that generates the entries of* T.

A general symmetric tridiagonal matrix T has $m = 2n-1$ degrees of freedom; however, $m < 2n-1$ is possible if the entries of T obey additional constraints (e.g., a zero main diagonal).

2.3. Perturbations and floating-point arithmetic.

In the following we often will have to consider the effect of perturbations on the eigenvalues (or singular values) and vectors.

Suppose a representation M of the matrix T is given by data δ_i. Then an *elementwise relative perturbation* (erp) of M to $\widetilde{\mathsf{M}}$ is defined by perturbing each δ_i to $\tilde{\delta}_i = \delta_i(1+\xi_i)$ with "small" $|\xi_i| \le \bar{\xi}$. To express this more compactly we will just write $\widetilde{\mathsf{M}} = \mathrm{erp}(\mathsf{M}, \bar{\xi})$, $\delta_i \rightsquigarrow \tilde{\delta}_i$, and although it must always be kept in mind that the perturbation applies to the data of the representation and not to the entries of T, we will sometimes write $\mathrm{erp}(\mathsf{T})$ for brevity.

A (partial) *relatively robust representation (RRR)* of a matrix T is one where small erps, bounded by some constant $\bar{\xi}$, in the data of the representation will cause only relative changes proportional to $\bar{\xi}$ in (some of) the eigenvalues and eigenvectors.

The need to consider perturbations comes from the rounding induced by computing in floating-point arithmetic. Throughout the paper we assume the standard model for floating-point arithmetic, namely that, barring underflow or overflow, the exact and computed results x and z of an arithmetic operation ($+$, $-$, $*$, $/$ and $\surd$) applied to floating-point numbers can be related as

$$x = z(1+\gamma) = z/(1+\delta), \quad |\gamma|, |\delta| \le \epsilon_\diamond,$$

with *machine epsilon* $\epsilon_\diamond$. For IEEE `double precision` with 53-bit significands and eleven-bit exponents we have $\epsilon_\diamond = 2^{-53} \approx 1.1 \cdot 10^{-16}$. For more information on binary floating-point arithmetic and the IEEE standard we refer the reader to [17, 23, 24, 26].

2.4. Eigenvalues and invariant subspaces.

The eigenvalues of a symmetric matrix A are real, and therefore they can be ordered ascendingly, $\lambda_1[\mathsf{A}] \le \ldots \le \lambda_n[\mathsf{A}]$, where the matrix will only be indicated if it is not clear from the context. The associated (orthonormal) eigenvectors are denoted by $\mathsf{q}_i[\mathsf{A}]$, and the invariant subspace spanned by a subset of the eigenvectors is $\mathcal{Q}_I[\mathsf{A}] := \mathrm{span}\{\mathsf{q}_i[\mathsf{A}] : i \in I\}$.

The sensitivity of the eigenvectors depends on the eigenvalue distribution—on the overall spread, measured by $\|\mathsf{A}\| = \max\{|\lambda_1|, |\lambda_n|\}$ or the *spectral diameter* $\mathrm{spdiam}[\mathsf{A}] = \lambda_n - \lambda_1$, as well as on the distance of an eigenvalue λ_i from the remainder of the spectrum. In a slightly more general form, the latter aspect is quantified by the notion of *gaps*, either in an absolute or a relative sense,

$$\begin{aligned} \mathrm{gap}_{\mathsf{A}}(I;\mu) &:= \min\big\{|\lambda_j - \mu| \,:\, j \notin I\big\}, \\ \mathrm{relgap}_{\mathsf{A}}(I) &:= \min\big\{|\lambda_j - \lambda_i|/|\lambda_i| \,:\, i \in I, j \notin I\big\}; \end{aligned}$$

see [37, Sect. 1]. Note that μ may, but need not, be an eigenvalue.

The following Gap Theorem [37, Thm. 2.1] is applied mostly in situations where I corresponds to a singleton ($|I| = 1$) or to a cluster of very close eigenvalues. The theorem states that if we have a "suspected eigenpair" (μ, x) with small residual, then x is indeed close to an eigenvector (or to the invariant subspace associated with the cluster) provided that μ is sufficiently far away from the remaining eigenvalues. For a formal definition of the (acute) angle see [37, Sect. 1].

THEOREM 2.2 (Gap Theorem for an invariant subspace). *For every symmetric matrix $\mathsf{A} \in \mathbb{R}^{n\times n}$, unit vector x, scalar μ and index set I, such that $\mathrm{gap}_{\mathsf{A}}(I;\mu) \neq 0$,*

$$\sin\angle\big(\mathsf{x}, \mathcal{Q}_I[\mathsf{A}]\big) \le \frac{\|\mathsf{A}\mathsf{x} - \mathsf{x}\mu\|}{\mathrm{gap}_{\mathsf{A}}(I;\mu)}.$$

For singletons, the Rayleigh quotient also provides a *lower* bound for the angle to an eigenvector.

THEOREM 2.3 (Gap Theorem with Rayleigh's quotient, [30, Thm. 11.7.1]). *For symmetric* $\mathsf{A} \in \mathbb{R}^{n\times n}$ *and unit vector* x *with* $\theta = \rho_{\mathsf{A}}(\mathsf{x}) := \mathsf{x}^*\mathsf{A}\mathsf{x}$*, let* $\lambda = \lambda_i[\mathsf{A}]$ *be an eigenvalue of* A *such that no other eigenvalue lies between (or equals)* λ *and* θ*, and* $\mathsf{q} = \mathsf{q}_i[\mathsf{A}]$ *the corresponding normalized eigenvector. Then we will have* $\mathrm{gap}_{\mathsf{A}}(\{i\};\theta) > 0$ *and*

$$\frac{\|\mathsf{A}\mathsf{x}-\theta\mathsf{x}\|}{\mathrm{spdiam}[\mathsf{A}]} \;\le\; \sin\angle\big(\mathsf{x},\mathsf{q}\big) \;\le\; \frac{\|\mathsf{A}\mathsf{x}-\theta\mathsf{x}\|}{\mathrm{gap}_{\mathsf{A}}(\{i\};\theta)} \qquad \textit{and} \qquad |\theta-\lambda| \;\le\; \frac{\|\mathsf{A}\mathsf{x}-\theta\mathsf{x}\|^2}{\mathrm{gap}_{\mathsf{A}}(\{i\};\theta)}.$$

2.5. Correctness of the MR^3 algorithm and requirements for proving it. In the analysis of the MR^3 algorithm in [37] the following five requirements have been identified, which together guarantee the correctness of Algorithm 2.1.

REQUIREMENT **RRR** (relatively robust representations). *There is a constant* C_{vecs} *such that for any perturbation* $\widetilde{\mathsf{M}} = \mathrm{erp}(\mathsf{M},\alpha)$ *at a node* (M, I)*, the effect on the eigenvectors can be controlled as*

$$\sin\angle\big(\mathcal{Q}_J[\mathsf{M}]\,,\,\mathcal{Q}_J[\widetilde{\mathsf{M}}]\big) \;\le\; C_{\mathrm{vecs}}\, n\alpha \big/ \mathrm{relgap}_{\mathsf{M}}(J),$$

for all $J \in \{I, I_1, \ldots, I_r\}$ *with* $|J| < n$*.*

This requirement also implies that singleton eigenvalues and the boundary eigenvalues of clusters cannot change by more than $\mathcal{O}(C_{\mathrm{vecs}} n\alpha|\lambda|)$ and therefore are relatively robust.

REQUIREMENT **ELG** (conditional element growth). *There is a constant* C_{elg} *such that for any perturbation* $\widetilde{\mathsf{M}} = \mathrm{erp}(\mathsf{M},\alpha)$ *at a node* (M, I)*, the incurred element growth is bounded by*

$$\begin{aligned} \|\widetilde{\mathsf{M}}-\mathsf{M}\| \;&\le\; \mathrm{spdiam}[\mathsf{M}_0], \\ \|(\widetilde{\mathsf{M}}-\mathsf{M})\bar{\mathsf{q}}_i\| \;&\le\; C_{\mathrm{elg}} n\alpha\, \mathrm{spdiam}[\mathsf{M}_0] \qquad \textit{for each } i \in I. \end{aligned}$$

This requirement concerns the *absolute* changes to matrix entries that result from *relative changes* to the representation data. For decomposition-based representations this is called *element growth (elg)*. Thus the requirement is fulfilled automatically if the matrix is represented by its entries directly. The two conditions convey that even large element growth is permissible (first condition), but only in those entries where the local eigenvectors of interest have tiny entries (second condition).

REQUIREMENT **RELGAPS** (relative gaps). *For each node* (M, I)*, the classification of* I *into child index sets in step 5 of Algorithm 2.1 is done such that for* $r = 1,\ldots,m$*,* $\mathrm{relgap}_{\mathsf{M}}(I_r) \ge gaptol$ *(if* $|I_r| < n$*).*

The parameter $gaptol$ is used to decide which eigenvalues are to be considered singletons and which ones are clustered. Typical values are $gaptol \sim 0.001\ldots 0.01$. Besides step 5, where fulfillment of the requirement should not be an issue if the eigenvalues are approximated accurately enough and the classification is done sensibly, this requirement also touches on the *outer* relative gaps of the whole local subset at the node. The requirement cannot be fulfilled if $\mathrm{relgap}_{\mathsf{M}}(I) < gaptol$. This fact has to be kept in mind when the node is created, in particular during evaluation of shifts for a new child in step 12.

REQUIREMENT **SHIFTREL** (shift relation). *There exist constants* $\alpha_{\downarrow}, \alpha_{\uparrow}$ *such that for every node with matrix* H *that was computed using shift* τ *as child of* M*, there are perturbations*

$$\breve{\mathsf{M}} = \mathrm{erp}(\mathsf{M},\alpha_{\downarrow}) \quad \textit{and} \quad \widehat{\mathsf{H}} = \mathrm{erp}(\mathsf{H},\alpha_{\uparrow})$$

with which the exact shift relation $\breve{\mathsf{M}} - \tau \;=\; \widehat{\mathsf{H}}$ *is attained.*

This requirement connects the nodes in the tree. It states that the computations of the shifted representations have to be done in a mixed relatively stable way. This is for example fulfilled when using twisted factorizations combined with qd-transformations as described in [8]. Improved variants of these techniques and a completely new approach based on block decompositions are presented in [35, 36, 38]. Note that the perturbation $\breve{\mathsf{M}} = \mathrm{erp}(\mathsf{M}, \alpha_\downarrow)$ at the parent will in general be different for each of its child nodes, but each child node has just one perturbation governed by $\alpha_\uparrow$ to establish the link to its parent node.

REQUIREMENT **GETVEC** (computation of eigenvectors). *There exist constants* $\alpha_\ddagger$, $\beta_\ddagger$ *and* R_{gv} *with the following property: Let* $(\bar\lambda^{\mathrm{leaf}}, \bar{\mathsf{q}})$ *with* $\bar{\mathsf{q}} = \bar{\mathsf{q}}_i$ *be computed at node* (M, I), *where* $\bar\lambda^{\mathrm{leaf}}$ *is the final local eigenvalue approximation. Then we can find elementwise perturbations to the matrix and the vector,*

$$\widetilde{\mathsf{M}} = \mathrm{erp}(\mathsf{M}, \alpha_\ddagger), \qquad \tilde{\mathsf{q}}(j) = \bar{\mathsf{q}}(j)(1 + \beta_j) \text{ with } |\beta_j| \le \beta_\ddagger,$$

for which the residual norm is bounded as

$$\big\|\mathsf{r}^{\mathrm{leaf}}\big\| \;:=\; \big\|(\widetilde{\mathsf{M}} - \bar\lambda^{\mathrm{leaf}})\tilde{\mathsf{q}}\big\| / \|\tilde{\mathsf{q}}\| \;\le\; R_{\mathrm{gv}} n \epsilon_\diamond \,\mathrm{gap}_{\widetilde{\mathsf{M}}}\big(\{i\}; \bar\lambda^{\mathrm{leaf}}\big).$$

This final requirement captures that the vectors computed in step 8 must have residual norms that are small, even when compared to the eigenvalue. The keys to fulfill this requirement are qd-type transformations to compute twisted factorizations $\mathsf{M} - \bar\lambda \;=:\; \mathsf{N}_k \mathsf{G}_k \mathsf{N}_k^*$ with mixed relative stability and then solving one of the systems $\mathsf{N}_k \mathsf{G}_k \mathsf{N}_k^* \bar{\mathsf{q}} = \gamma_k \mathsf{e}_k$ for the eigenvector [8, 12, 31].

In practice, we expect the constants C_{vecs} and C_{elg} to be of moderate size (~ 10), $\alpha_\downarrow$, $\alpha_\uparrow$, and $\alpha_\ddagger$ should be $\mathcal{O}(\epsilon_\diamond)$, whereas $\beta_\ddagger = \mathcal{O}(n\epsilon_\diamond)$, and R_{gv} may become as large as $\mathcal{O}(1/\mathit{gaptol})$. Thus the following theorems provide bounds $\mathit{resid}_{\mathsf{M}_0} = \mathcal{O}(n\epsilon_\diamond \|\mathsf{M}_0\| / \mathit{gaptol})$ for the residuals and $\mathit{orth}_{\mathsf{M}_0} = \mathcal{O}(n\epsilon_\diamond / \mathit{gaptol})$ for the orthogonality.

THEOREM 2.4 (Residual norms for MR^3 [37, Thm. 3.1]). *Let the representation tree traversed by Algorithm 2.1 satisfy the requirements* ELG, SHIFTREL, *and* GETVEC. *For given index* $j \in I_0$, *let* $d = \mathrm{depth}(j)$ *be the depth of the node where* $\bar{\mathsf{q}} = \bar{\mathsf{q}}_j$ *was computed (cf. Figure 2.1) and* $\mathsf{M}_0, \mathsf{M}_1, \ldots, \mathsf{M}_d$ *be the representations along the path from the root* (M_0, I_0) *to that node, with shifts* τ_i *linking* M_i *and* M_{i+1}, *respectively. Then*

$$\big\|(\mathsf{M}_0 - \lambda^*)\bar{\mathsf{q}}\big\| \;\le\; \Big(\big\|\mathsf{r}^{\mathrm{leaf}}\big\| + \gamma\,\mathrm{spdiam}[\mathsf{M}_0]\Big)\,\frac{1+\beta_\ddagger}{1-\beta_\ddagger} \;=:\; \mathit{resid}_{\mathsf{M}_0},$$

where $\lambda^* := \tau_0 + \cdots + \tau_{d-1} + \bar\lambda^{\mathrm{leaf}}$ *and* $\gamma := C_{\mathrm{elg}}\, n\,\big(d(\alpha_\downarrow + \alpha_\uparrow) + \alpha_\ddagger\big) + 2(d+1)\beta_\ddagger$.

The following theorem confirms the orthogonality of the computed eigenvectors and bounds their angles to the local invariant subspaces. It combines Lemma 3.4 and Theorem 3.5 from [37].

THEOREM 2.5. *Let the representation tree traversed by Algorithm 2.1 fulfill the requirements* RRR, RELGAPS, SHIFTREL, *and* GETVEC. *Then for each node* (M, I) *in the tree with child index set* $J \subseteq I$, *the computed vectors* $\bar{\mathsf{q}}_j$, $j \in J$, *will obey*

$$\sin\angle\big(\bar{\mathsf{q}}_j\,,\,\mathcal{Q}_J[\mathsf{M}]\big) \;\le\; C_{\mathrm{vecs}}\big(\alpha_\ddagger + (\mathrm{depth}(j) - \mathrm{depth}(\mathsf{M}))(\alpha_\downarrow + \alpha_\uparrow)\big)n/\mathit{gaptol} + \kappa,$$

where $\kappa := R_{\mathrm{gv}} n \epsilon_\diamond + \beta_\ddagger$. *Moreover, any two computed vectors* $\bar{\mathsf{q}}_i$ *and* $\bar{\mathsf{q}}_j$, $i \ne j$, *will obey*

$$\tfrac{1}{2}\bar{\mathsf{q}}_i^* \bar{\mathsf{q}}_j \;\le\; C_{\mathrm{vecs}}\big(\alpha_\ddagger + \mathrm{d}_{\max}(\alpha_\downarrow + \alpha_\uparrow)\big)n/\mathit{gaptol} + \kappa \;=:\; \mathit{orth}_{\mathsf{M}_0},$$

where $\mathrm{d}_{\max} := \max\{\mathrm{depth}(i) \,|\, i \in I_0\}$ *denotes the maximum depth of a node in the tree.*

3. The singular value decomposition of bidiagonal matrices.

In this section we briefly review the problem BSVD and its close connection to the eigenvalue problem for tridiagonal symmetric matrices.

3.1. The problem.

Throughout this paper we consider $\mathsf{B} \in \mathbb{R}^{n\times n}$, an upper bidiagonal matrix with diagonal entries a_i and offdiagonal elements b_i, that is,

$$\mathsf{B} \;=\; \mathrm{diag}(a_1,\ldots,a_n) + \mathrm{diag}_{+1}(b_1,\ldots,b_{n-1}).$$

The goal is to compute the full *singular value decomposition*

$$\mathsf{B} \;=\; \mathsf{U}\Sigma\mathsf{V}^* \quad \text{with} \quad \mathsf{U}^*\mathsf{U} = \mathsf{V}^*\mathsf{V} = \mathsf{I},\ \Sigma = \mathrm{diag}(\sigma_1,\ldots,\sigma_n),\ \text{and } \sigma_1 \le \cdots \le \sigma_n. \tag{3.1}$$

The columns $\mathsf{u}_i = \mathsf{U}(:,i)$ and $\mathsf{v}_i = \mathsf{V}(:,i)$ are called *left* and *right singular vectors*, respectively, and the σ_i are the *singular values*. Taken together, $(\sigma_i, \mathsf{u}_i, \mathsf{v}_i)$ form a *singular triplet* of B. Note that we order the singular values *ascendingly* in order to simplify the transition between BSVD and TSEP.

For any algorithm solving BSVD, the computed singular triplets $(\bar{\sigma}_i, \bar{\mathsf{u}}_i, \bar{\mathsf{v}}_i)$ should be *numerically orthogonal* in the sense

$$\max\left\{|\bar{\mathsf{U}}^*\bar{\mathsf{U}} - \mathsf{I}|, |\bar{\mathsf{V}}^*\bar{\mathsf{V}} - \mathsf{I}|\right\} \;=\; \mathcal{O}(n\epsilon_\diamond), \tag{3.2}$$

where $|\cdot|$ is to be understood componentwise. We also desire small *residual norms*,

$$\max_i\left\{\|\mathsf{B}\bar{\mathsf{v}}_i - \bar{\mathsf{u}}_i\bar{\sigma}_i\|, \|\mathsf{B}^*\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_i\bar{\sigma}_i\|\right\} \;=\; \mathcal{O}(\|\mathsf{B}\|n\epsilon_\diamond). \tag{3.3}$$

In the literature the latter is sometimes stated as the singular vector pairs being “(well) coupled.”

3.2. Singular values to high relative accuracy.

In [4] Demmel and Kahan established that every bidiagonal matrix (represented by entries) determines its singular values to high relative accuracy.

The current state-of-the-art for computing singular values is the dqds-algorithm by Fernando and Parlett [14, 32], which builds upon [4] as well as Rutishauser’s original qd-algorithm [34]. An excellent implementation of dqds is included in LAPACK in the form of routine `xLASQ1`. Alternatively, bisection could be used, but this is normally much slower—in our experience it becomes worthwhile to use bisection instead of dqds only if less than ten percent of the singular values are desired (dqds can only be used to compute all singular values).

The condition (3.3) alone does merely convey that each computed $\bar{\sigma}_i$ must lie within distance $\mathcal{O}(\|\mathsf{B}\|n\epsilon_\diamond)$ of *some* exact singular value of B. A careful but elementary argument based on the Gap Theorem 2.2 (applied to the Golub–Kahan matrix, see below) shows that (3.2) and (3.3) combined actually provide for *absolute accuracy* in the singular values, meaning each computed $\bar{\sigma}_i$ lies within distance $\mathcal{O}(\|\mathsf{B}\|n\epsilon_\diamond)$ of the exact σ_i. To achieve *relative accuracy*, a straightforward modification is just to recompute the singular values afterwards using, for example, dqds. It is clear that doing so cannot spoil (3.3), at least as long as $\bar{\sigma}_i$ was computed with absolute accuracy. The recomputation does not even necessarily be overhead; for MR^3-type algorithms like those we study in this paper one needs initial approximations to the singular values anyway, the more accurate the better. So there is actually a gain from computing them up front to full precision.

3.3. Associated tridiagonal problems.

There are two standard approaches to reduce the problem BSVD to TSEP, involving three different symmetric tridiagonal matrices.

3.3.1. The normal equations. From (3.1) we can see the eigendecompositions of the symmetric tridiagonal matrices BB^* and $\mathsf{B}^*\mathsf{B}$ to be

$$\mathsf{BB}^* = \mathsf{U}\Sigma^2\mathsf{U}^*, \quad \mathsf{B}^*\mathsf{B} = \mathsf{V}\Sigma^2\mathsf{V}^*.$$

These two are called *normal equations*, analogously to the linear least squares problem. The individual entries of BB^* and $\mathsf{B}^*\mathsf{B}$ can be expressed using those of B:

$$\begin{aligned}\mathsf{BB}^* &= \operatorname{diag}\big(a_1^2+b_1^2,\ldots,a_{n-1}^2+b_{n-1}^2,a_n^2\big)+\operatorname{diag}_{\pm1}\big(a_2b_1,\ldots,a_nb_{n-1}\big),\\ \mathsf{B}^*\mathsf{B} &= \operatorname{diag}\big(a_1^2,a_2^2+b_1^2,\ldots,a_n^2+b_{n-1}^2\big)+\operatorname{diag}_{\pm1}\big(a_1b_1,\ldots,a_{n-1}b_{n-1}\big).\end{aligned}$$

Arguably the most straightforward approach to tackle the BSVD would be to just employ the MR^3 algorithm for TSEP (Algorithm 2.1) to compute eigendecompositions of BB^* and $\mathsf{B}^*\mathsf{B}$ separately. This gives both left and right singular vectors as well as the singular values (twice). A slight variation on this theme would compute just the vectors on one side, for example $\mathsf{BB}^* = \mathsf{U}\Sigma^2\mathsf{U}^*$, and then get the rest through solving $\mathsf{Bv} = \mathsf{u}\sigma$. As BB^* and $\mathsf{B}^*\mathsf{B}$ are already positive definite bidiagonal factorizations, we would naturally take them directly as root representations, avoiding the mistake to form either matrix product explicitly.

In short, this black box approach is a bad idea. While the matrices $\bar{\mathsf{U}}$ and $\bar{\mathsf{V}}$ computed via the two TSEPs are orthogonal almost to working precision, the residuals $\|\mathsf{B}\bar{\mathsf{v}}_i - \bar{\mathsf{u}}_i\bar{\sigma}_i\|$ and $\|\mathsf{B}^*\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_i\bar{\sigma}_i\|$ may be $\mathcal{O}(\sigma_i)$ for clustered singular values, which is unacceptable for large σ_i. Roughly speaking, this comes from computing $\bar{\mathsf{U}}$ and $\bar{\mathsf{V}}$ independently – so there is no guarantee that the corresponding $\bar{\mathsf{u}}_i$ and $\bar{\mathsf{v}}_i$ "fit together." Note that this problem is not tied to taking MR^3 as eigensolver but also occurs if QR or divide and conquer are used to solve the two TSEPs *independently*.

With MR^3 it is, however, possible to "couple" the solution of the two TSEPs in a way that allows to control the residuals. This is done by running MR^3 on only one of the matrices BB^* or $\mathsf{B}^*\mathsf{B}$, say BB^*, and "simulating" the action of MR^3 on $\mathsf{B}^*\mathsf{B}$ with the same sequence of shifts, that is, with an identical representation tree; cf. Figure 2.1. The key to this strategy is the observation that the quantities that would be computed in MR^3 on $\mathsf{B}^*\mathsf{B}$ can also be obtained from the respective quantities in the BB^*-run via so-called *coupling relations*. For several reasons the Golub–Kahan matrix (see the following discussion) is also involved in the couplings. See [19, 20, 21, 39] for the development of the coupling approach and [35] for a substantially revised version.

In our experiments, however, an approach based entirely on the Golub–Kahan matrix turned out to be superior, and therefore we will not pursue the normal equations and the coupling approach further in the current paper.

3.3.2. The Golub–Kahan matrix. Given an upper bidiagonal matrix B we obtain a symmetric eigenproblem of twice the size by forming the *Golub–Kahan (GK) matrix* or *Golub–Kahan form* of B [13],

$$\mathsf{T}_{\text{GK}}(\mathsf{B}) := \mathsf{P}_{\text{ps}}\begin{bmatrix}0 & \mathsf{B}\\ \mathsf{B}^* & 0\end{bmatrix}\mathsf{P}_{\text{ps}}^*,$$

where P_{ps} is the *perfect shuffle* permutation on $\mathbb{R}^{2n}$ that maps any $\mathsf{x}\in\mathbb{R}^{2n}$ to

$$\mathsf{P}_{\text{ps}}\mathsf{x} = \big[\mathsf{x}(n+1),\mathsf{x}(1),\mathsf{x}(n+2),\mathsf{x}(2),\ldots,\mathsf{x}(2n),\mathsf{x}(n)\big]^*,$$

or, equivalently stated,

$$\mathsf{P}_{\text{ps}}^*\mathsf{x} = \big[\mathsf{x}(2),\mathsf{x}(4),\ldots,\mathsf{x}(2n),\mathsf{x}(1),\mathsf{x}(3),\ldots,\mathsf{x}(2n-1)\big]^*.$$

It is easy to verify that $\mathsf{T}_{\rm GK}(\mathsf{B})$ is a symmetric tridiagonal matrix with a zero diagonal and the entries of B interleaved on the offdiagonals,

$$\mathsf{T}_{\rm GK}(\mathsf{B}) \;=\; \operatorname{diag}_{\pm1}(a_1, b_1, a_2, b_2, \ldots, a_{n-1}, b_{n-1}, a_n),$$

and that its eigenpairs are related to the singular triplets of B via

$$\begin{aligned} & (\sigma, \mathsf{u}, \mathsf{v}) \text{ is a singular triplet of } \mathsf{B} \text{ with } \|\mathsf{u}\| = \|\mathsf{v}\| = 1 \\ \text{iff}\quad & (\pm\sigma, \mathsf{q}) \text{ are eigenpairs or } \mathsf{T}_{\rm GK}(\mathsf{B}),\ \text{ where } \|\mathsf{q}\| = 1, \mathsf{q} = \tfrac{1}{\sqrt{2}}\mathsf{P}_{\rm ps}\begin{bmatrix}\mathsf{u}\\ \pm\mathsf{v}\end{bmatrix}. \end{aligned}$$

Thus v makes up the odd-numbered entries in q and u the even-numbered ones:

$$\mathsf{q} \;=\; \frac{1}{\sqrt{2}}\big[\mathsf{v}(1), \mathsf{u}(1), \mathsf{v}(2), \mathsf{u}(2), \ldots, \mathsf{v}(n), \mathsf{u}(n)\big]^*. \tag{3.4}$$

It will frequently be necessary to relate rotations of GK eigenvectors q to rotations of their u and v components. This is captured in the following lemma. The formulation has been kept fairly general; in particular the permutation $\mathsf{P}_{\rm ps}$ is left out, but the claim does extend naturally if it is reintroduced.

LEMMA 3.1. *Let q, q' be non-orthogonal unit vectors that admit a conforming partition*

$$\mathsf{q} = \begin{bmatrix}\mathsf{u}\\ \mathsf{v}\end{bmatrix}, \quad \mathsf{q}' = \begin{bmatrix}\mathsf{u}'\\ \mathsf{v}'\end{bmatrix}, \quad \mathsf{u}, \mathsf{v} \neq \mathsf{o}.$$

Let $\varphi_{\mathsf{u}} := \angle(\mathsf{u}, \mathsf{u}')$, $\varphi_{\mathsf{v}} := \angle(\mathsf{v}, \mathsf{v}')$ and $\varphi := \angle(\mathsf{q}, \mathsf{q}')$. Then

$$\begin{aligned} \max\Big\{\|\mathsf{u}\|\sin\varphi_{\mathsf{u}}, \|\mathsf{v}\|\sin\varphi_{\mathsf{v}}\Big\} \;&\le\; \sin\varphi, \\ \max\Big\{\big|\|\mathsf{u}'\| - \|\mathsf{u}\|\big|, \big|\|\mathsf{v}'\| - \|\mathsf{v}\|\big|\Big\} \;&\le\; \frac{\sin\varphi + (1-\cos\varphi)}{\cos\varphi}. \end{aligned}$$

Proof. Define r such that

$$\mathsf{q} \;=\; \begin{bmatrix}\mathsf{u}\\ \mathsf{v}\end{bmatrix} \;=\; \mathsf{q}'\cos\varphi + \mathsf{r} \;=\; \begin{bmatrix}\mathsf{u}'\cos\varphi + \mathsf{r}_{\mathsf{u}}\\ \mathsf{v}'\cos\varphi + \mathsf{r}_{\mathsf{v}}\end{bmatrix}.$$

The resulting situation is depicted in Figure 3.1. Consequently,

$$\|\mathsf{u}\|\sin\varphi_{\mathsf{u}} \;\le\; \|\mathsf{r}_{\mathsf{u}}\| \;\le\; \|\mathsf{r}\| \;=\; \sin\varphi.$$

Now $\mathsf{u}'\cos\varphi = \mathsf{u} - \mathsf{r}_{\mathsf{u}}$ implies $(\mathsf{u}' - \mathsf{u})\cos\varphi = (1-\cos\varphi)\mathsf{u} - \mathsf{r}_{\mathsf{u}}$. Use the reverse triangle inequality and $\|\mathsf{u}\| < 1$ for

$$\begin{aligned} \big|\|\mathsf{u}'\| - \|\mathsf{u}\|\big|\cos\varphi \;&\le\; \|(\mathsf{u}'-\mathsf{u})\cos\varphi\| = \|(1-\cos\varphi)\mathsf{u} - \mathsf{r}_{\mathsf{u}}\| \le (1-\cos\varphi)\|\mathsf{u}\| + \|\mathsf{r}_{\mathsf{u}}\| \\ &\le\; (1-\cos\varphi) + \sin\varphi \end{aligned}$$

and divide by $\cos\varphi \neq 0$ to obtain the desired bound for $\big|\|\mathsf{u}'\| - \|\mathsf{u}\|\big|$. The claims pertaining to the v components are shown analogously. □

Application to a given approximation q' for an exact GK eigenvector q merely requires to exploit $\|\mathsf{u}\| = \|\mathsf{v}\| = 1/\sqrt{2}$. In particular, the second claim of Lemma 3.1 will then enable us to control how much the norms of u' and v' can deviate from $1/\sqrt{2}$, namely basically by no more than $\sin\varphi + \mathcal{O}(\sin^2\varphi)$, provided φ is small, which will be the case in later applications. (For large φ, the bound in the lemma may be larger than the obvious $\max\big\{\big|\|\mathsf{u}'\| - \|\mathsf{u}\|\big|, \big|\|\mathsf{v}'\| - \|\mathsf{v}\|\big|\big\} \le 1$, given that all these vectors have length at most 1.)

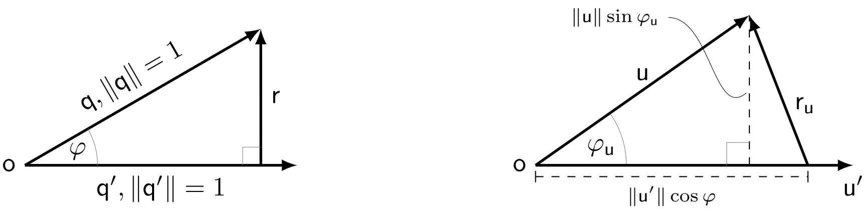

FIG. 3.1. *Situation for the proof of Lemma 3.1. The global setting is on the left, the right side zooms in just on the* u *components. Note that in general* $\varphi_{\mathsf{u}} \neq \varphi$ *and* r_{u} *will not be orthogonal to* u*, nor to* u′.

3.4. Preprocessing. Before actually solving the BSVD problem, the given input matrix B should be preprocessed with regard to some points. In contrast to TSEP, where it suffices to deal with the offdiagonal elements, now all entries of B are involved with the offdiagonals of $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})$, which makes preprocessing a bit more difficult.

If the input matrix is lower bidiagonal, work with B^* instead and swap the roles of U and V. Multiplication on both sides by suitable diagonal signature matrices makes all entries nonnegative, and we can scale to get the largest elements into proper range. Then, in order to avoid several numerical problems later on, it is highly advisable to get rid of tiny entries by setting them to zero and splitting the problem. To summarize, we should arrive at

$$n\epsilon_\diamond \|\mathsf{B}\| \; < \; \min\{a_i, b_i\}. \tag{3.5}$$

However, splitting a bidiagonal matrix to attain (3.5) by setting all violating entries to zero is not straightforward. Two issues must be addressed.

If an offdiagonal element b_i is zero, B is reducible and can be partitioned into two smaller bidiagonal problems. If a diagonal element a_i is zero then B is singular. An elegant way to "deflate" one zero singular value is to apply one sweep of the implicit zero-shift QR method, which will yield a matrix B′ with $b'_{i-1} = b'_{n-1} = a'_n = 0$, cf. [4, p. 21]. Thus the zero singular value has been revealed and can now be removed by splitting into three upper bidiagonal parts $\mathsf{B}_{1:i-1}$, $\mathsf{B}_{i:n-1}$ and $\mathsf{B}_{n,n}$, the latter of which is trivial. An additional benefit of the QR sweep is a possible preconditioning effect for the problem [19], but of course we will also have to rotate the computed vectors afterwards.

The second obstacle is that using (3.5) as criterion for setting entries to zero will impede computing the singular values to high relative accuracy with respect to the input matrix. There are splitting criteria which retain relative accuracy, for instance those employed within the zero-shift QR algorithm [4, p. 18] and the slightly stronger ones by Li [28, 32]. However, all these criteria allow for less splitting than (3.5).

To get the best of both, that is, extensive splitting with all its benefits as well as relatively accurate singular values, we propose a 2-phase splitting as follows:

1) Split the matrix as much as possible *without* spoiling relative accuracy. This results in a partition of B into blocks $\mathsf{B}_{\mathrm{rs}}^{(1)}, \ldots, \mathsf{B}_{\mathrm{rs}}^{(N)}$, which we call the *relative split* of B.
2) Split each block $\mathsf{B}_{\mathrm{rs}}^{(i)}$ further aggressively into blocks $\mathsf{B}_{\mathrm{as}}^{(i,1)}, \ldots, \mathsf{B}_{\mathrm{as}}^{(i,n_i)}$ to achieve (3.5). We denote the collection of subblocks $\mathsf{B}_{\mathrm{as}}^{(i,j)}$ as *absolute split* of B.
3) Solve BSVD for each block in the absolute split independently.
4) Use bisection to refine the computed singular values of each block $\mathsf{B}_{\mathrm{as}}^{(i,j)}$ to high relative accuracy with respect to the parent block $\mathsf{B}_{\mathrm{rs}}^{(i)}$ in the relative split.

Since the singular values of the blocks in the absolute split retain absolute accuracy with respect to B, the requirements (3.2) and (3.3) will still be upheld. In fact, if dqds is used to precompute the singular values (cf. Section 3.2) one can even skip steps 1) and 4), since the singular values that are computed for the blocks of the absolute split are discarded anyways. The sole purpose of the separate relative split is to speed up the refinement in step 4).

We want to stress that we propose the 2-phase splitting also when only a subset of singular triplets is desired. Then an additional obstacle is to get a consistent mapping of triplet indices between the blocks. This can be done efficiently, but it is not entirely trivial.

4. MR^3 and the Golub–Kahan matrix.

In this section we investigate the approach to use MR^3 on the Golub–Kahan matrix to solve the problem BSVD.

A *black box* approach would employ MR^3 "as is," without modifications to its internals, to compute eigenpairs of $\mathsf{T}_{\text{GK}}(\mathsf{B})$ and then extract the singular vectors via (3.4). Here the ability of MR^3 to compute partial spectra is helpful, as we need only concern ourselves with one half of the spectrum of $\mathsf{T}_{\text{GK}}(\mathsf{B})$. Note that using MR^3 this way would also offer to compute only a subset of singular triplets at reduced cost; current solution methods for BSVD like divide-and-conquer or QR do not provide this feature.

The standing opinion for several years has been that there are fundamental problems involved which cannot be overcome, in particular concerning the orthogonality of the extracted left and right singular vectors. The main objective of this section is to refute that notion.

We start our exposition with a numerical experiment to indicate that using MR^3 as a pure black box method on the Golub–Kahan matrix is indeed not a sound idea.

EXAMPLE 4.1. We used LAPACK's test matrix generator `DLATMS` to construct a bidiagonal matrix with the following singular values, ranging between $0.9 \cdot 10^{-8}$ and 110.

$$\begin{aligned}
\sigma_{13} &= 0.9, \quad \sigma_{14} = 1-10^{-7}, \quad \sigma_{15} = 1+10^{-7}, \quad \sigma_{16} = 1.1,\\
\sigma_i &= \sigma_{i+4}/100, \quad i = 12, 11, \ldots, 1,\\
\sigma_i &= 100\cdot\sigma_{i-4}, \quad i = 17, \ldots, 20.
\end{aligned}$$

Then we formed the symmetric tridiagonal matrix $\mathsf{T}_{\text{GK}}(\mathsf{B}) \in \mathbb{R}^{40\times 40}$ explicitly. The MR^3 implementation `DSTEMR` from LAPACK 3.2.1 was called to give us the upper 20 eigenpairs $(\bar\sigma_i, \bar{\mathsf{q}}_i)$ of $\mathsf{T}_{\text{GK}}(\mathsf{B})$. The matrix is well within numerical range, so that `DSTEMR` neither splits nor scales the tridiagonal problem. The singular vectors were then extracted via

$$\begin{bmatrix}\bar{\mathsf{u}}_i\\ \bar{\mathsf{v}}_i\end{bmatrix} := \sqrt{2}\mathsf{P}_{\text{ps}}^*\bar{\mathsf{q}}_i.$$

The results are shown in Figure 4.1. The left plot clearly shows that `DSTEMR` does its job of solving the eigenproblem posed by $\mathsf{T}_{\text{GK}}(\mathsf{B})$. But the right plot conveys just as clearly that the extracted singular vectors are far from being orthogonal. In particular, the small singular values are causing trouble. Furthermore, the u and v components have somehow lost their property of having equal norm. However, their norms are still close enough to one that normalizing them explicitly would not improve the orthogonality levels significantly.

This experiment is not special—similar behavior can be observed consistently for other test cases with small singular values. The explanation is simple: MR^3 does neither know, nor care, what a Golub–Kahan matrix is. It will start just as always, by first choosing a shift outside the spectrum, say $\tau \lesssim -\sigma_n$, and compute $\mathsf{T}_{\text{GK}}(\mathsf{B}) - \tau = \mathsf{L}_0\mathsf{D}_0\mathsf{L}_0^*$ as positive definite root representation. From there it will then deploy further shifts into the spectrum of $\mathsf{L}_0\mathsf{D}_0\mathsf{L}_0^*$ to isolate the requested eigenpairs.

What happens is that the first shift to the outside smears all small singular values into one cluster, as shown in Figure 4.2. Consider for instance we have $\|\mathsf{B}\| \geq 1$ and are working with the standard $gaptol = 0.001$. We can even assume the initial shift was done exactly; so let $\lambda_{\pm i}^{(0)} = \sigma_{\pm i} - \tau$ be the eigenvalues of $\mathsf{L}_0\mathsf{D}_0\mathsf{L}_0^*$. Then for all indices i with $\sigma_i \lesssim 0.0005$ the

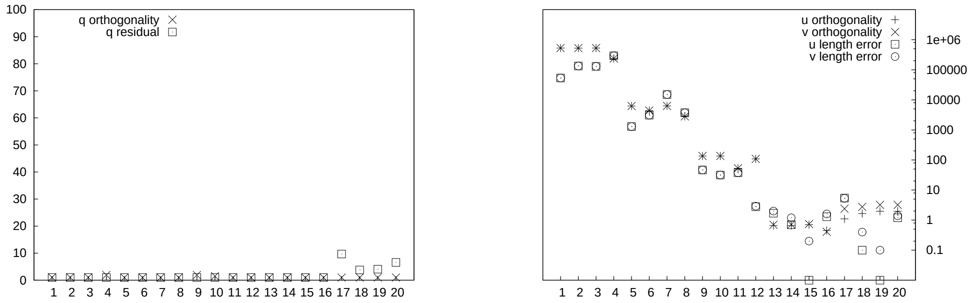

FIG. 4.1. *Data for Example 4.1, on a per-vector basis,* $i = 1, \dots, 20$. *Left: scaled orthogonality* $\|\bar{\mathsf{Q}}^*\bar{\mathsf{q}}_i - \mathsf{e}_i\|_\infty / n\epsilon_\diamond$ *with* $\mathsf{e}_i = (0, \dots, 0, 1, 0, \dots, 0)^*$ *denoting the* i*-th unit vector, and scaled residuals* $\|\mathsf{T}_{\mathrm{GK}}(\mathsf{B})\bar{\mathsf{q}}_i - \bar{\mathsf{q}}_i\bar{\sigma}_i\| / 2\|\mathsf{B}\| n\epsilon_\diamond$ *for* TSEP. *Right: scaled orthogonality* $\|\mathsf{U}^*\bar{\mathsf{u}}_i - \mathsf{e}_i\|_\infty / n\epsilon_\diamond$, $\|\mathsf{V}^*\bar{\mathsf{v}}_i - \mathsf{e}_i\|_\infty / n\epsilon_\diamond$ *and scaled deviation from unit length,* $\big|\|\bar{\mathsf{u}}_i\|^2 - 1\big| / n\epsilon_\diamond$, $\big|\|\bar{\mathsf{v}}_i\|^2 - 1\big| / n\epsilon_\diamond$, *for* BSVD.

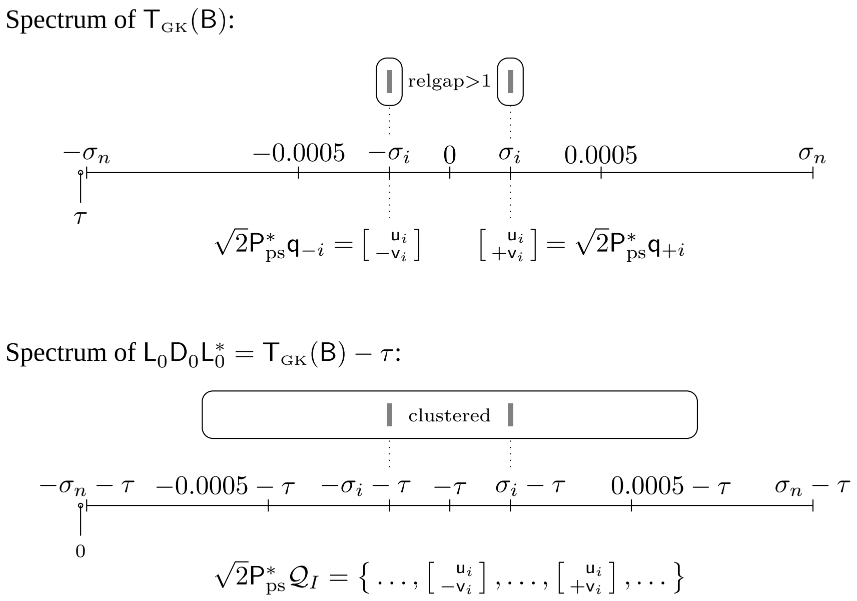
Spectrum of $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})$:
relgap>1
$-\sigma_n$
-0.0005
$-\sigma_i$
0
σ_i
0.0005
σ_n
τ
$\sqrt{2}\mathsf{P}_{\mathrm{ps}}^*\mathsf{q}_{-i} = \left[\begin{smallmatrix}\mathsf{u}_i\\ -\mathsf{v}_i\end{smallmatrix}\right]$
$\left[\begin{smallmatrix}\mathsf{u}_i\\ +\mathsf{v}_i\end{smallmatrix}\right] = \sqrt{2}\mathsf{P}_{\mathrm{ps}}^*\mathsf{q}_{+i}$
Spectrum of $\mathsf{L}_0\mathsf{D}_0\mathsf{L}_0^* = \mathsf{T}_{\mathrm{GK}}(\mathsf{B}) - \tau$:
clustered
$-\sigma_n - \tau$
$-0.0005 - \tau$
$-\sigma_i - \tau$
$-\tau$
$\sigma_i - \tau$
$0.0005 - \tau$
$\sigma_n - \tau$
0
$\sqrt{2}\mathsf{P}_{\mathrm{ps}}^*\mathcal{Q}_I = \left\{\dots, \left[\begin{smallmatrix}\mathsf{u}_i\\ -\mathsf{v}_i\end{smallmatrix}\right], \dots, \left[\begin{smallmatrix}\mathsf{u}_i\\ +\mathsf{v}_i\end{smallmatrix}\right], \dots\right\}$

FIG. 4.2. *Why the naive black box approach of MR*^{3} *on* T_{GK} *is doomed.*

corresponding $\lambda_{\pm i}^{(0)}$ will belong to the same cluster of $\mathsf{L}_0\mathsf{D}_0\mathsf{L}_0^*$, since their *relative* distance is

$$\frac{|\lambda_{+i}^{(0)} - \lambda_{-i}^{(0)}|}{\max\left\{|\lambda_{+i}^{(0)}|, |\lambda_{-i}^{(0)}|\right\}} = \frac{(\sigma_i - \tau) - (-\sigma_i - \tau)}{\sigma_i - \tau} = \frac{2\sigma_i}{\sigma_i - \tau} < gaptol.$$

Therefore, for such a singular triplet $(\sigma_i, \mathsf{u}_i, \mathsf{v}_i)$ of B, *both* of $\mathsf{P}_{\mathrm{ps}}\left[\begin{smallmatrix}\mathsf{u}_i\\ \pm\mathsf{v}_i\end{smallmatrix}\right]$ will be eigenvectors associated with that cluster of $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})$. Hence, further (inexact) shifts based on this configuration cannot guarantee to separate them again cleanly. Consequently, using MR^3 as black box on the Golub–Kahan matrix in this fashion could in principle even produce eigenvectors q with *identical* u or v components.

This problem is easy to overcome. After all we know that the entries of $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})$ form an RRR, so the initial outside shift to find a positive definite root representation is completely

```
Input:   Upper bidiagonal $\mathsf{B} \in \mathbb{R}^{n\times n}$, index set $I_0 \subseteq \{1,\dots,n\}$
Output:  Singular triplets $(\bar\sigma_i, \bar{\mathsf{u}}_i, \bar{\mathsf{v}}_i), i \in I_0$

1.  Execute the MR$^3$ algorithm for TSEP (Algorithm 2.1), but take $\mathsf{M}_0 := \mathsf{T}_{\mathrm{GK}}(\mathsf{B})$ as root
    representation in step 1, using the entries of $\mathsf{B}$ directly.
    This gives eigenpairs $(\bar\sigma_i, \bar{\mathsf{q}}_i), i \in I_0$.
2.  Extract the singular vectors via $\begin{bmatrix}\bar{\mathsf{u}}_i\\ \bar{\mathsf{v}}_i\end{bmatrix} := \sqrt{2}\mathsf{P}_{\mathrm{ps}}^*\bar{\mathsf{q}}_i$.
```

Algorithm 4.1: MR^3 on the Golub–Kahan matrix. Compute specified singular triplets of bidiagonal B using the MR^3 algorithm on $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})$.

unnecessary—we can just take $\mathsf{M}_0 := \mathsf{T}_{\mathrm{GK}}(\mathsf{B})$ directly as root. For shifting, that is, for computing a child representation $\mathsf{M}^+ = \mathsf{T}_{\mathrm{GK}}(\mathsf{B}) - \mu$ on the first level, a special routine exploiting the zero diagonal should be employed. If M^+ is to be a twisted factorization this is much easier to do than standard `dtwqds`; see [13, 25] and our remarks in [38, Sect. 8.3]. With this setting, small singular values can be handled by a (positive) shift in one step, without danger of spoiling them by unwanted contributions from the negative counterparts. This solution method is sketched in Algorithm 4.1. Note that we now have heterogeneous representation types in the tree, as the root $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})$ is represented by its entries. In any case, our general setup of MR^3 and its proof in [35, 37] can handle this situation.

One can argue that the approach is still flawed on a fundamental level. Großer gives an example in [19] which we want to repeat at this point. In fact his argument can be fielded against using *any* TSEP-solver on the Golub–Kahan matrix for BSVD.

EXAMPLE 4.2 (cf. Beispiel 1.33 in [19]). Assume the exact GK eigenvectors

$$\mathsf{P}_{\mathrm{ps}}^*\mathsf{q}_i = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathsf{u}_i\\ \mathsf{v}_i\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\\1\\1\\-1\end{bmatrix}, \quad \mathsf{P}_{\mathrm{ps}}^*\mathsf{q}_j = \frac{1}{\sqrt{2}}\begin{bmatrix}\mathsf{u}_j\\ \mathsf{v}_j\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\\-1\\1\\1\end{bmatrix},$$

form (part of) the basis for a cluster. The computed vectors will generally not be exact, but might for instance be $\mathsf{G}_{\mathrm{rot}}\mathsf{P}_{\mathrm{ps}}^*\big[\mathsf{q}_i \,|\, \mathsf{q}_j\big]$, where $\mathsf{G}_{\mathrm{rot}}$ is a rotation $\left[\begin{smallmatrix}c & s\\ -s & c\end{smallmatrix}\right]$, $c^2 + s^2 = 1$, in the 2-3 plane. We end up with computed singular vectors

$$\sqrt{2}\bar{\mathsf{u}}_i = \begin{bmatrix}1\\ c+s\end{bmatrix}, \sqrt{2}\bar{\mathsf{u}}_j = \begin{bmatrix}1\\ s-c\end{bmatrix}, \quad \sqrt{2}\bar{\mathsf{v}}_i = \begin{bmatrix}c-s\\ -1\end{bmatrix}, \sqrt{2}\bar{\mathsf{v}}_j = \begin{bmatrix}c+s\\ 1\end{bmatrix},$$

that have orthogonality levels $|\mathsf{u}_i^*\mathsf{u}_j| = |\mathsf{v}_i^*\mathsf{v}_j| = s^2$.

However, this rotation does leave the invariant subspace spanned by q_i and q_j (cf. Lemma 4.4 below), so if s^2 is large, the residual norms of $\bar{\mathsf{q}}_i$ and $\bar{\mathsf{q}}_j$ would suffer, too.

That the extracted singular vectors can be far from orthogonal even if the GK vectors are fine led Großer to the conclusion that there must be a fundamental problem. Until recently we believed that as well [39, p. 914]. However, we will now set out to prove that with just a small additional requirement, Algorithm 4.1 will actually work. This is a new result and shows that there is no *fundamental* problem in using MR^3 on the Golub–Kahan matrix. Of particular interest is that the situation in Example 4.2—which, as we mentioned, would apply to all TSEP solvers on T_{GK}—can be avoided if MR^3 is deployed as in Algorithm 4.1.

The following definition will let us control the danger that the shifts within MR^3 lose information about the singular vectors.

DEFINITION 4.3. *A subspace $\mathcal{S}$ of $\mathbb{R}^{2n\times 2n}$ with orthonormal basis $(\mathsf{q}_i)_{i\in I}$ is said to have* GK structure *if the systems $(\mathsf{u}_i)_{i\in I}$ and $(\mathsf{v}_i)_{i\in I}$ of vectors extracted according to*

$$\begin{bmatrix}\mathsf{u}_i\\ \mathsf{v}_i\end{bmatrix} := \sqrt{2}\mathsf{P}_{\mathrm{ps}}^{*}\mathsf{q}_i, \quad i\in I,$$

are orthonormal each.

The special property of a GK matrix is that all invariant subspaces belonging to (at most) the first or second half of the spectrum have GK structure. As eigenvectors are shift-invariant, this property carries over to any matrix that can be written as $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})-\mu$ for suitable B, which is just any symmetric tridiagonal matrix of even dimension with a *constant diagonal*.

The next lemma reveals that the u and v components of every vector within a subspace with GK structure have equal norm. Thus the actual choice of the orthonormal system (q_i) in Definition 4.3 is irrelevant.

LEMMA 4.4. *Let the subspace $\mathcal{S}\subseteq\mathbb{R}^{2n\times 2n}$ have GK structure. Then for each $\mathsf{s}\in\mathcal{S}$,*

$$\sqrt{2}\mathsf{s} \;=\; \mathsf{P}_{\mathrm{ps}}\begin{bmatrix}\mathsf{s}_u\\ \mathsf{s}_v\end{bmatrix} \quad \textit{with} \quad \|\mathsf{s}_u\|=\|\mathsf{s}_v\|.$$

Proof. As $\mathcal{S}$ has GK structure, we have an orthonormal basis $(\mathsf{q}_1,\ldots,\mathsf{q}_m)$ for $\mathcal{S}$ such that

$$\sqrt{2}\mathsf{P}_{\mathrm{ps}}^{*}\mathsf{q}_i \;=\; \begin{bmatrix}\mathsf{u}_i\\ \mathsf{v}_i\end{bmatrix}, \quad i=1,\ldots,m,$$

with orthonormal u_i and v_i. Each $s\in\mathcal{S}$ can be written as $\mathsf{s}=\alpha_1\mathsf{q}_1+\cdots+\alpha_m\mathsf{q}_m$, and therefore

$$\sqrt{2}\mathsf{P}_{\mathrm{ps}}^{*}\mathsf{s} = \begin{bmatrix}\alpha_1\mathsf{u}_1+\cdots+\alpha_m\mathsf{u}_m\\ \alpha_1\mathsf{v}_1+\cdots+\alpha_m\mathsf{v}_m\end{bmatrix} =: \begin{bmatrix}\mathsf{s}_{\mathsf{u}}\\ \mathsf{s}_{\mathsf{v}}\end{bmatrix}.$$

Since the u_i and v_j are orthonormal we have $\|\mathsf{s}_{\mathsf{u}}\|^2=\sum\alpha_i^2=\|\mathsf{s}_{\mathsf{v}}\|^2$. □

Now comes the proof of concrete error bounds for Algorithm 4.1. The additional requirement we need is that the local subspaces are kept "near" to GK structure. We will discuss how to handle this requirement in practice afterwards.

For simplicity we assume that the call to MR^3 in step 1 of Algorithm 4.1 produces perfectly normalized vectors, $\|\bar{\mathsf{q}}_i\| = 1$, and that the multiplication by $\sqrt{2}$ in step 2 is done exactly.

THEOREM 4.5 (Proof of correctness for Algorithm 4.1). *Let Algorithm 4.1 be executed such that the representation tree built by MR^3 satisfies all five requirements listed in Section 2.5. Furthermore, let each node (M,I) have the property that a suitable perturbation $\widetilde{\mathsf{M}}_{\mathrm{GK}} = \mathrm{erp}(\mathsf{M},\xi_{\mathrm{GK}})$ can be found such that the subspace $\mathcal{Q}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]$ has GK structure. Finally, let $resid_{\mathrm{GK}}$ and $orth_{\mathrm{GK}}$ denote the right-hand side bounds from Theorem 2.4 and from the second inequality in Theorem 2.5, respectively. Then the computed singular triplets will satisfy*

$$\begin{aligned}
\max\left\{\cos\angle(\bar{\mathsf{u}}_i,\bar{\mathsf{u}}_j),\cos\angle(\bar{\mathsf{v}}_i,\bar{\mathsf{v}}_j)\right\} &\leq 2\sqrt{2}A, \quad i\neq j,\\
\max\left\{\big|\|\bar{\mathsf{u}}_i\|-1\big|,\big|\|\bar{\mathsf{v}}_i\|-1\big|\right\} &\leq \sqrt{2}A+\mathcal{O}(A^2),\\
\max\left\{\|\mathsf{B}\bar{\mathsf{v}}_i-\bar{\mathsf{u}}_i\bar{\sigma}_i\|,\|\mathsf{B}^{*}\bar{\mathsf{u}}_i-\bar{\mathsf{v}}_i\bar{\sigma}_i\|\right\} &\leq \sqrt{2}\,resid_{\mathrm{GK}},
\end{aligned}$$

where $A := orth_{\mathrm{GK}} + C_{\mathrm{vecs}}n\xi_{\mathrm{GK}}/gaptol.$

Proof. As all requirements for MR^3 are fulfilled, Theorems 2.4 and 2.5 apply for the computed GK eigenpairs $(\bar\sigma_i, \bar{\mathsf{q}}_i)$.

We will first deal with the third bound concerning the residual norms. Invoke the definition of the Golub–Kahan matrix to see

$$\mathsf{T}_{\mathrm{GK}}(\mathsf{B})\bar{\mathsf{q}}_i - \bar{\mathsf{q}}_i\bar\sigma_i \;=\; \tfrac{1}{\sqrt{2}}\mathsf{P}_{\mathrm{ps}}\begin{bmatrix}\mathsf{B}\bar{\mathsf{v}}_i - \bar{\mathsf{u}}_i\bar\sigma_i\\ \mathsf{B}^*\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_i\bar\sigma_i\end{bmatrix}$$

and then use Theorem 2.4 to obtain

$$\|\mathsf{B}\bar{\mathsf{v}}_i - \bar{\mathsf{u}}_i\bar\sigma_i\|^2 + \|\mathsf{B}^*\bar{\mathsf{u}}_i - \bar{\mathsf{v}}_i\bar\sigma_i\|^2 \;=\; 2\|\mathsf{T}_{\mathrm{GK}}(\mathsf{B})\bar{\mathsf{q}}_i - \bar{\mathsf{q}}_i\bar\sigma_i\|^2 \;\le\; 2\mathit{resid}_{\mathrm{GK}}^2.$$

For orthogonality, consider indices i and j and let (M, N) be the last common ancestor of i and j, i.e., the deepest node in the tree such that $i \in I$ and $j \in J$ for different child index sets $I, J \subseteq N$. The bound $\mathit{orth}_{\mathrm{GK}}$ on the right-hand side in the second inequality in Theorem 2.5 is just the worst-case for the first inequality in that theorem, taken over all nodes in the tree. Hence we have

$$\sin\angle\big(\bar{\mathsf{q}}_i\,,\,\mathcal{Q}_I[\mathsf{M}]\big) \;\le\; \mathit{orth}_{\mathrm{GK}}.$$

As we assume that the representation M fulfills Requirements RRR and RELGAPS, we can link $\bar{\mathsf{q}}_i$ to the nearby matrix $\widetilde{\mathsf{M}}_{\mathrm{GK}}$ by

$$\begin{aligned}\sin\angle\big(\bar{\mathsf{q}}_i\,,\,\mathcal{Q}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]\big) &\le \sin\angle\big(\bar{\mathsf{q}}_i\,,\,\mathcal{Q}_I[\mathsf{M}]\big) + \sin\angle\big(\mathcal{Q}_I[\mathsf{M}]\,,\,\mathcal{Q}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]\big)\\ &\le \mathit{orth}_{\mathrm{GK}} + C_{\mathrm{vecs}}n\xi_{\mathrm{GK}}\big/\mathit{gaptol} = A.\end{aligned}$$

This means we can find a unit vector $\mathsf{q} \in \mathcal{Q}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]$ with $\sin\angle\big(\bar{\mathsf{q}}_i, \mathsf{q}\big) \le A$.

Now $\mathcal{Q}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}] \subseteq \mathcal{Q}_N[\widetilde{\mathsf{M}}_{\mathrm{GK}}]$ has GK structure. By Lemma 4.4 we can therefore partition

$$\sqrt{2}\mathsf{q} \;=\; \mathsf{P}_{\mathrm{ps}}\begin{bmatrix}\mathsf{u}\\ \mathsf{v}\end{bmatrix} \quad \text{with } \|\mathsf{u}\| = \|\mathsf{v}\| = 1.$$

Let $\mathcal{U}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]$ denote the subspace spanned by the u components of vectors in $\mathcal{Q}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]$. Thus $\mathsf{u} \in \mathcal{U}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]$, and Lemma 3.1 gives

$$\sin\angle\big(\bar{\mathsf{u}}_i\,,\,\mathcal{U}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]\big) \;\le\; \sin\angle\big(\bar{\mathsf{u}}_i, \mathsf{u}\big) \;\le\; \sqrt{2}A,$$

as well as the desired property $\big|\|\bar{\mathsf{u}}_i\| - 1\big| \le \sqrt{2}A + \mathcal{O}(A^2)$ for the norms. Repeat the steps above for j to arrive at $\sin\angle\big(\bar{\mathsf{u}}_j\,,\,\mathcal{U}_J[\widetilde{\mathsf{M}}_{\mathrm{GK}}]\big) \;\le\; \sqrt{2}A$. We can write

$$\begin{aligned}\bar{\mathsf{u}}_i &= \mathsf{x} + \mathsf{r}, \quad \mathsf{x} \in \mathcal{U}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}], \quad \mathsf{r} \perp \mathsf{x}, \quad \|\mathsf{r}\| = \|\bar{\mathsf{u}}_i\|\sin\angle\big(\bar{\mathsf{u}}_i\,,\,\mathcal{U}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]\big),\\ \bar{\mathsf{u}}_j &= \mathsf{y} + \mathsf{s}, \quad \mathsf{y} \in \mathcal{U}_J[\widetilde{\mathsf{M}}_{\mathrm{GK}}], \quad \mathsf{s} \perp \mathsf{y}, \quad \|\mathsf{s}\| = \|\bar{\mathsf{u}}_j\|\sin\angle\big(\bar{\mathsf{u}}_j\,,\,\mathcal{U}_J[\widetilde{\mathsf{M}}_{\mathrm{GK}}]\big).\end{aligned}$$

Since $\mathcal{Q}_N[\widetilde{\mathsf{M}}_{\mathrm{GK}}]$ has GK structure and $I \cap J = \emptyset$, the spaces $\mathcal{U}_I[\widetilde{\mathsf{M}}_{\mathrm{GK}}]$ and $\mathcal{U}_J[\widetilde{\mathsf{M}}_{\mathrm{GK}}]$ are orthogonal, and in particular $\mathsf{x} \perp \mathsf{y}$. Therefore

$$|\bar{\mathsf{u}}_i^*\bar{\mathsf{u}}_j| \;=\; |\mathsf{x}^*(\mathsf{y}+\mathsf{s}) + \mathsf{r}^*\bar{\mathsf{u}}_j| \;\le\; |\mathsf{x}^*\mathsf{s}| + |\mathsf{r}^*\bar{\mathsf{u}}_j| \;\le\; \|\mathsf{x}\|\|\mathsf{s}\| + \|\mathsf{r}\|\|\bar{\mathsf{u}}_j\|,$$

where we made use of $\mathsf{x}^*\mathsf{y} = 0$ for the first inequality and invoked the Cauchy–Schwartz inequality for the second one. Together with $\|\mathsf{x}\| \le \|\bar{\mathsf{u}}_i\|$, this yields

$$\cos\angle\big(\bar{\mathsf{u}}_i, \bar{\mathsf{u}}_j\big) \;=\; \frac{|\bar{\mathsf{u}}_i^*\bar{\mathsf{u}}_j|}{\|\bar{\mathsf{u}}_i\|\,\|\bar{\mathsf{u}}_j\|} \;\le\; \frac{\|\mathsf{s}\|}{\|\bar{\mathsf{u}}_j\|} + \frac{\|\mathsf{r}\|}{\|\bar{\mathsf{u}}_i\|} \;\le\; 2\sqrt{2}A.$$

The bounds for the right singular vectors v_i are obtained analogously. □

One conclusion from Theorem 4.5 is that it really does not matter if we extract the singular vectors as done in step 2 of Algorithm 4.1 by multiplying the q subvectors by $\sqrt{2}$, or if we normalize them explicitly.

The new requirement that was introduced in Theorem 4.5 is stated minimally, namely that the representations M can be perturbed to yield local invariant subspaces with GK structure. In this situation we say that the subspace of M "nearly" has GK structure. At the moment we do not see a way to specifically test for this property. However, we do know that any even-dimensioned symmetric tridiagonal matrix with a constant diagonal is just a shifted Golub–Kahan matrix, so trivially each subspace (within one half) has GK structure. Let us capture this.

DEFINITION 4.6. *If for a given representation of symmetric tridiagonal* M *there exists an elementwise relative perturbation*

$$\widetilde{\mathsf{M}} \;=\; \mathrm{erp}(\mathsf{M},\xi) \quad \textit{such that} \quad \widetilde{\mathsf{M}}(i,i) \;\equiv\; c,$$

then we say that M *has a* nearly constant diagonal, *in short* M is ncd, *or, if more detail is to be conveyed,* $\mathsf{M} \in \mathrm{ncd}(c)$ *or* $\mathsf{M} \in \mathrm{ncd}(c,\xi)$.

Clearly, the additional requirement for Theorem 4.5 is fulfilled if all representations in the tree are ncd. Note that a representation being ncd does not necessarily imply that all diagonal entries are about equal, because there might be large local element growth. For example, LDL^* can be ncd even if $|d_i| \gg |(\mathsf{LDL}^*)(i,i)|$ for some index i, cf. Example 4.8 below.

The ncd property can easily and cheaply be verified in practice, e.g., for an LDL^* factorization with the condition $|(\mathsf{LDL}^*)(i,i) - \text{const}| = \mathcal{O}(\epsilon_\diamond)\cdot\max\{\,|d_i|, |\ell_{i-1}^2 d_{i-1}|\,\}$ for all $i > 1$. Note that the successively shifted descendants of a Golub–Kahan matrix can only violate the ncd property if there was large local element growth at some diagonal entries on the way.

REMARK 4.7. Since Theorem 4.5 needs the requirement SHIFTREL anyway, the shifts $\mathsf{T}_{\mathrm{GK}}(\mathsf{B}) - \mu = \mathsf{M}^+$ to get to level one must be executed with mixed relative stability. Therefore, all representations on level one will automatically be ncd and as such always fulfill the extra requirement of having subspaces near to GK structure, independent of element growth or relative condition numbers.

The preceding theoretical results will be demonstrated in action by numerical experiments in Section 5. Those will confirm that Algorithm 4.1 is indeed a valid solution strategy for BSVD. However, it will also become apparent that working with a Golub–Kahan matrix as root can sometimes be problematic in practice. The reason is that Golub–Kahan matrices are highly vulnerable to element growth when confronted with a tiny shift.

EXAMPLE 4.8. (Cf. [36, Example 1.2]) Let $\alpha \ll 1$ (e.g., $\alpha \sim \epsilon_\diamond$) and consider the bidiagonal matrix $\mathsf{B} = \begin{bmatrix} 1 & 1 \\ & \alpha \end{bmatrix}$ with singular values $\sigma_1 \approx \alpha$, $\sigma_2 \approx 2$. Shifting $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})$ by $-\alpha$ gives

$$\begin{bmatrix} -\alpha & 1 & & \\ 1 & -\alpha & 1 & \\ & 1 & -\alpha & \alpha \\ & & \alpha & -\alpha \end{bmatrix} = \mathsf{LDL}^*$$

with $\mathsf{D} = \mathrm{diag}\big(-\alpha,\ \frac{1-\alpha^2}{\alpha},\ -\alpha\frac{2-\alpha^2}{1-\alpha^2},\ -\alpha\frac{1}{2-\alpha^2}\big)$. Clearly there is huge local element growth in $\mathsf{D}(2)$. This LDL^* still is ncd, but if we had to shift it again the property would probably be lost completely.

The thing is that we really have no way to avoid a tiny shift if clusters of tiny singular values are present. In [35, 36] a generalization to twisted factorizations called *block factorizations* is investigated. The latter are especially suited for shifting Golub–Kahan matrices and essentially render the above concerns obsolete.

5. Numerical results. In this section we present the results that were obtained with our prototype implementation of Algorithm 4.1, XMR-TGK, on two test sets Pract and Synth. We also compare to XMR-CPL, which implements the coupling approach for running MR^3 on the normal equations; cf. Section 3.3.1.

Most of the bidiagonal matrices in the test sets were obtained from tridiagonal problems T in two steps: (1) T was scaled and split to enforce $e_i > \epsilon_\diamond \|\mathsf{T}\|, \quad i = 1, \ldots, n-1$. (2) For each unreduced subproblem we chose a shift to allow a Cholesky decomposition, yielding an upper bidiagonal matrix.

The Pract test set contains 75 bidiagonal matrices with dimensions up to 6245. They were obtained in the above manner from tridiagonal matrices from various applications. For more information about the specific matrices see [5], where the same set was used to evaluate the symmetric eigensolvers in LAPACK.

The Synth set contains 19240 bidiagonal matrices that stem from artificially generated tridiagonal problems, including standard types like Wilkinson matrices as well as matrices with eigenvalue distributions built into LAPACK's test matrix generator DLATMS. In fact, all artificial types listed in [29] are present.

For each of these basic types, all tridiagonal matrices up to dimension 100 were generated. Then these were split according to step (1) above. For the resulting tridiagonal subproblems we made two further versions by gluing [9, 33] them to themselves: either two copies with a small glue $\gtrsim \|\mathsf{T}\| n\epsilon_\diamond$ or three copies with two medium $\mathcal{O}(\|\mathsf{T}\| n\sqrt{\epsilon_\diamond})$ glues. Finally, step (2) above was used to obtain bidiagonal factors of all unreduced tridiagonal matrices.

Further additions to Synth include some special bidiagonal matrices B that were originally devised by Benedikt Großer. These were glued as well. However, special care was taken that step (1) above would not affect the matrix $\mathsf{B}^*\mathsf{B}$ for any one of these extra additions.

The code XMR-TGK is based on a *prototype* MR^3 TSEP solver, XMR, which essentially implements Algorithm 4.1. XMR differs from the LAPACK implementation DSTEMR mainly in the following points.

- DSTEMR relies on twisted factorizations $\mathsf{T} = \mathsf{N}_k \mathsf{G}_k \mathsf{N}_k^*$, represented by the nontrivial entries $d_1, \ldots, d_{k-1}, \gamma_k, r_{k+1}, \ldots, r_n$ from the matrix G_k in (2.1) and the offdiagonal entries $\ell_1, \ldots, \ell_{k-1}, u_{k+1}, \ldots, u_n$ from N_k, whereas XMR uses the same entries from G_k, together with the $n-1$ offdiagonals $\ell_1 d_1, \ldots, \ell_{k-1} d_{k-1}, u_{k+1} r_{k+1}, \ldots, u_n r_n$ of the *tridiagonal* matrix T. This "e–representation" provides somewhat smaller error bounds at comparable cost; see [35, 38] for more details.
- Even if the relative robustness (Requirement RRR) and moderate element growth (Requirement ELG) cannot always be guaranteed before actually performing a shift, sufficient a priori criteria are available. These have been improved in XMR.
- Several other modifications have been incorporated to enhance robustness and efficiency, e.g., in the interplay of Rayleigh quotient iteration and bisection, and in the bisection strategy.

An optimized *production* implementation of XMR is described in [37].

XMR-TGK adapts the tridiagonal XMR to the BSVD by using $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})$ as root representation. To cushion the effect of moderate element growth on the diagonal we also switched to using "Z–representations" for the children nodes. These representations again use the entries of G_k, together with the $n-1$ quantities $\ell_1^2 d_1, \ldots, \ell_{k-1}^2 d_{k-1}, u_{k+1}^2 r_{k+1}, \ldots, u_n^2 r_n$, and they provide even sharper error bounds, albeit at higher cost; cf. [35, 38]. In addition to the checks in XMR, a shift candidate has to be $\mathrm{ncd}(-\bar{\mu}, 32n\epsilon_\diamond)$ in order to be considered acceptable a priori; see the discussion following Theorem 4.5.

As the coupled approach is not discussed in the present paper, we can only briefly hint at the main features of the implementation XMR-CPL; see [35] for more details. XMR-CPL essentially performs XMR on the Golub–Kahan matrix ("*central layer*") and uses coupling relations to implicitly run the MR^3 algorithm simultaneously on the matrices $\mathsf{B}\mathsf{B}^*$ and $\mathsf{B}^*\mathsf{B}$ as well ("*outer layers*"). Just like XMR-TGK we use Z–representations in the central layer, and the representations there have to fulfill the same ncd-condition, but the other a priori acceptance conditions in XMR are only checked for the outer representations. Eigenvalue refinements are done on the side that gives the better a priori bound for relative condition. To counter the fact that for the coupled approach we cannot prove that SHIFTREL holds always, appropriate consistency checks with Sturm counts are done for *both* outer representations.

Table 5.1 summarizes the orthogonality levels and residual norms of XMR-TGK and XMR-CPL on the test sets. XMR-TGK works amazingly well. Indeed, the extracted vectors have better orthogonality than what LAPACK's implementation DSTEMR provides for $\mathsf{B}^*\mathsf{B}$ alone, and they are not much worse than those delivered by XMR.

The coupled approach works also well on Pract, but has some undeniable problems with Synth. Indeed, not shown in the tables is that for 24 of the cases in Synth, XMR-CPL failed to produce up to 2.04% of the singular triplets. The reason is that for those cases there were clusters where none of the tried shift candidates satisfied the aforementioned consistency checks for the child eigenvalue bounds to replace the missing SHIFTREL. Note that these failures are not errors, since the code did flag the triplets as not computed.

Finally let us consider the matrix from Example 4.1, which yields unsatisfactory orthogonality with a "black box" MR^3 on the Golub–Kahan matrix (see Example 4.1) and large residuals with black box MR^3 on the normal equations (not shown in this paper). By contrast, both XMR-TGK and XMR-CPL solve this problem with worst orthogonality levels of $1.15n\epsilon_\diamond$ and BSVD-residual norms $0.68\|\mathsf{B}\|n\epsilon_\diamond$. Interestingly these two numbers are identical for both methods, whereas the computed vectors differ.

The accuracy results would mean that the coupled approach is clearly outclassed by using MR^3 on the Golub–Kahan matrix in the fashion of Algorithm 4.1, if it were not for efficiency. Counting the subroutine calls reveals that XMR-CPL does more bisections (for checking the couplings) and more RQI steps (to compute the second vector), but these operations are on size-n matrices, whereas the matrices in XMR-TGK all have size $2n$. Thus we expect XMR-CPL to perform about $20-30\%$ faster than XMR-TGK.

These results give in fact rise to a third method for BSVD, namely a combination of the first two: Use MR^3 on the Golub–Kahan matrix $\mathsf{T}_{\mathrm{GK}}(\mathsf{B})$ like in Algorithm 4.1, but employ the coupling relations to outsource the expensive eigenvalue refinements to smaller matrices of half the size. This approach would retain the increased accuracy of XMR-TGK at reduced cost, without the need for coupling checks. The catch is that we still need the "central layer" (translates of T_{GK}) to be robust for XMR-TGK, but to do the eigenvalue computations with one "outer layer" (translates of $\mathsf{B}\mathsf{B}^*$ or $\mathsf{B}^*\mathsf{B}$) the representation there has to be robust as well. This would be a consequence of Theorem 5.2 in [21], but its proof contains a subtle error. The combined method is new and sounds promising, in particular if *block factorizations* (introduced in [35, 36]) are used to increase the accuracy. At the moment we favor XMR-TGK because it leads to a much leaner implementation and can profit directly from any improvement in the underlying tridiagonal MR^3 algorithm.

Acknowledgments. The authors want to thank Osni Marques and Christof Vömel for providing them with the Pract test matrices and the referees for their helpful suggestions.

TABLE 5.1

Comparison of the orthogonality levels $\max\{|\mathsf{U}^*\mathsf{U}-\mathsf{I}|,|\mathsf{V}^*\mathsf{V}-\mathsf{I}|\}/n\epsilon_\diamond$ *and the residual norms* $\max_i\{\|\mathsf{B}\bar{\mathsf{v}}_i-\bar{\mathsf{u}}_i\bar{\sigma}_i\|,\|\mathsf{B}^*\bar{\mathsf{u}}_i-\bar{\mathsf{v}}_i\bar{\sigma}_i\|\}/\|\mathsf{B}\|n\epsilon_\diamond$ *of* `XMR-TGK` *and* `XMR-CPL`*. The lines below* MAX *give the percentages of test cases with maximum residual and loss of orthogonality, respectively, in the indicated ranges.*

Pract (75 cases)			**Synth** (19240 cases)	
`XMR-TGK`	`XMR-CPL`		`XMR-TGK`	`XMR-CPL`
Orthogonality level $\max\{|\mathsf{U}^*\mathsf{U}-\mathsf{I}|,|\mathsf{V}^*\mathsf{V}-\mathsf{I}|\}\ /\ n\epsilon_\diamond$				
5.35	10.71	AVG	5.34	6.33
2.71	2.44	MED	1.38	1.01
48.40	154	MAX	3095	27729
81.33 %	82.67 %	$0\ldots10$	92.59 %	91.04 %
18.67 %	14.67 %	$10\ldots100$	7.04 %	8.61 %
	2.67 %	$100\ldots200$	0.12 %	0.21 %
		$200\ldots500$	0.11 %	0.10 %
		$500\ldots10^3$	0.07 %	0.02 %
		$10^3\ldots10^6$	0.06 %	0.03 %
Residual norms $\max_i\{\|\mathsf{B}\bar{\mathsf{v}}_i-\bar{\mathsf{u}}_i\bar{\sigma}_i\|,\|\mathsf{B}^*\bar{\mathsf{u}}_i-\bar{\mathsf{v}}_i\bar{\sigma}_i\|\}\ /\ \|\mathsf{B}\|n\epsilon_\diamond$				
0.35	15.78	AVG	0.45	3.14
0.07	1.37	MED	0.13	0.72
4.19	453	MAX	118	6873
92.00 %	34.67 %	$0\ldots1$	84.96 %	57.45 %
8.00 %	50.67 %	$1\ldots10$	15.03 %	35.50 %
	8.00 %	$10\ldots100$		7.00 %
	6.67 %	>100	0.01 %	0.06 %

REFERENCES

[1] E. ANDERSON, Z. BAI, C. H. BISCHOF, L. S. BLACKFORD, J. W. DEMMEL, J. J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. J. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.

[2] C. H. BISCHOF, B. LANG, AND X. SUN, *A framework for symmetric band reduction*, ACM Trans. Math. Software, 26 (2000), pp. 581–601.

[3] J. J. M. CUPPEN, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.

[4] J. W. DEMMEL AND W. KAHAN, *Accurate singular values of bidiagonal matrices*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 873–912.

[5] J. W. DEMMEL, O. A. MARQUES, B. N. PARLETT, AND C. VÖMEL, *Performance and accuracy of LAPACK's symmetric tridiagonal eigensolvers*, SIAM J. Sci. Comput., 30 (2008), pp. 1508–1526.

[6] I. S. DHILLON, *A new $O(n^2)$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem*, Ph.D. Thesis, Computer Science Division, University of California, Berkeley, 1997.

[7] I. S. DHILLON AND B. N. PARLETT, *Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices*, Linear Algebra Appl., 387 (2004), pp. 1–28.

[8] ———, *Orthogonal eigenvectors and relative gaps*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 858–899.

[9] I. S. DHILLON, B. N. PARLETT, AND C. VÖMEL, *Glued matrices and the MRRR algorithm*, SIAM J. Sci. Comput., 27 (2005), pp. 496–510.

[10] Z. DRMAČ AND K. VESELIĆ, *New fast and accurate Jacobi SVD algorithm I.*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1322–1342.

[11] ———, *New fast and accurate Jacobi SVD algorithm II.*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1343–1362.

[12] K. V. FERNANDO, *On computing an eigenvector of a tridiagonal matrix. I. Basic results*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 1013–1034.

[13] ———, *Accurately counting singular values of bidiagonal matrices and eigenvalues of skew-symmetric tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 373–399.

[14] K. V. FERNANDO AND B. N. PARLETT, *Accurate singular values and differential qd algorithms*, Numer. Math., 67 (1994), pp. 191–229.

[15] J. G. F. FRANCIS, *The QR transformation: a unitary analogue to the LR transformation I.*, Comput. J., 4 (1961/62), pp. 265–272.

[16] ———, *The QR transformation II.*, Comput. J., 4 (1961/62), pp. 332–345.

[17] D. GOLDBERG, *What every computer scientist should know about floating-point arithmetic*, ACM Computing Surveys, 23 (1991), pp. 5–48.

[18] G. H. GOLUB AND W. KAHAN, *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2 (1965), pp. 205–224.

[19] B. GROSSER, *Ein paralleler und hochgenauer $O(n^2)$ Algorithmus für die bidiagonale Singulärwertzerlegung*, Ph.D. Thesis, Fachbereich Mathematik, Bergische Universität Gesamthochschule Wuppertal, Wuppertal, Germany, 2001.

[20] B. GROSSER AND B. LANG, *An $O(n^2)$ algorithm for the bidiagonal SVD*, Linear Algebra Appl., 358 (2003), pp. 45–70.

[21] ———, *On symmetric eigenproblems induced by the bidiagonal SVD*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 599–620.

[22] M. GU AND S. C. EISENSTAT, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 172–191.

[23] IEEE, *IEEE Standard 754-1985 for Binary Floating-Point Arithmetic*, Aug. 1985.

[24] ———, *IEEE Standard 754-2008 for Floating-Point Arithmetic*, Aug. 2008.

[25] W. KAHAN, *Accurate eigenvalues of a symmetric tri-diagonal matrix*, Tech. Report CS41, Computer Science Department, Stanford University, July 1966.

[26] ———, *Lecture notes on the status of IEEE standard 754 for binary floating point arithmetic*, 1995. http://www.cs.berkeley.edu/∼wkahan/ieee754status/IEEE754.PDF.

[27] B. LANG, *Reduction of banded matrices to bidiagonal form*, Z. Angew. Math. Mech., 76 (1996), pp. 155–158.

[28] R.-C. LI, *Relative perturbation theory. I. Eigenvalue and singular value variations*, SIAM J. Matrix Anal. Appl., 19 (1998), pp. 956–982.

[29] O. A. MARQUES, C. VÖMEL, J. W. DEMMEL, AND B. N. PARLETT, *Algorithm 880: a testing infrastructure for symmetric tridiagonal eigensolvers*, ACM Trans. Math. Software, 35 (2008), pp. 8:1–8:13.

[30] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice Hall, Englewood Cliffs, NJ, 1980.

[31] B. N. PARLETT AND I. S. DHILLON, *Fernando's solution to Wilkinson's problem: an application of double factorization*, Linear Algebra Appl., 267 (1997), pp. 247–279.

[32] B. N. PARLETT AND O. A. MARQUES, *An implementation of the dqds algorithm (positive case)*, Linear Algebra Appl., 309 (2000), pp. 217–259.

[33] B. N. PARLETT AND C. VÖMEL, *The spectrum of a glued matrix*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 114–132.

[34] H. RUTISHAUSER, *Der Quotienten-Differenzen-Algorithmus*, Z. Angew. Math. Phys., 5 (1954), pp. 233–251.

[35] P. R. WILLEMS, *On MR^3-type Algorithms for the Tridiagonal Symmetric Eigenproblem and the Bidiagonal SVD*, Ph.D. Thesis, Fachbereich Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, Wuppertal, Germany, 2010.

[36] P. R. WILLEMS AND B. LANG, *Block factorizations and qd-type transformations for the MR^3 algorithm*, Electron. Trans. Numer. Anal., 38 (2011), pp. 363–400. http://etna.math.kent.edu/vol.38.2011/pp363-400.dir.

[37] ———, *A framework for the MR^3 algorithm: theory and implementation*, Preprint BUW-SC 2011/2, Fachbereich Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, Wuppertal, Germany, 2011.

[38] ———, *Twisted factorizations and qd-type transformations for the MR^3 algorithm—new representations and analysis*, Preprint BUW-SC 2011/3, Fachbereich Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, Wuppertal, Germany, 2011.

[39] P. R. WILLEMS, B. LANG, AND C. VÖMEL, *Computing the bidiagonal SVD using multiple relatively robust representations*, SIAM J. Matrix Anal. Appl., 28 (2007), pp. 907–926.