Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Linear Algebra and its Applications 387 (2004) 1–28

LINEAR ALGEBRA AND ITS APPLICATIONS

www.elsevier.com/locate/laa

Multiple representations to compute orthogonal eigenvectors of symmetric tridiagonal matrices

Inderjit S. Dhillon [a,1], Beresford N. Parlett [b,*]

[a]Department of Computer Science, University of Texas, Austin, TX 78712-1188, USA
[b]Mathematics Department and Computer Science Division, EECS Department, University of California, Berkeley, CA 94720, USA

Received 1 August 2001; accepted 10 December 2003

Submitted by J.L. Barlow

Abstract

In this paper we present an $O(nk)$ procedure, Algorithm MR^3, for computing k eigenvectors of an $n \times n$ symmetric tridiagonal matrix T. A salient feature of the algorithm is that a number of different LDL^t products (L unit lower triangular, D diagonal) are computed. In exact arithmetic each LDL^t is a factorization of a translate of T. We call the various LDL^t products *representations* (of T) and, roughly speaking, there is a representation for each cluster of close eigenvalues. The unfolding of the algorithm, for each matrix, is well described by a *representation tree*. We present the tree and use it to show that if each representation satisfies three prescribed conditions then the computed eigenvectors are orthogonal to working accuracy and have small residual norms with respect to the original matrix T.

Keywords: Symmetric tridiagonal; Eigenvectors; Orthogonality; High relative accuracy; Relatively robust representations (RRR)

1. Introduction

In this paper, we present an algorithm that takes a real $n \times n$ symmetric tridiagonal matrix and computes approximate eigenvectors that are orthogonal to working accuracy, under prescribed conditions. We call our method Algorithm MR^3 or MRRR

[*] Corresponding author.

E-mail addresses: inderjit@cs.utexas.edu (I.S. Dhillon), parlett@math.berkeley.edu (B.N. Parlett).

[1] The author's research was supported by NSF CAREER Award no. ACI-0093404 and a startup grant from the University of Texas, Austin.

0024-3795/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.laa.2003.12.028

(Algorithm of Multiple Relatively Robust Representations) for reasons that will become clear. To compute k eigenvectors, this algorithm requires $\mathrm{O}(nk)$ operations in the worst case whereas previous algorithms such as the QR algorithm, the Divide and Conquer method and inverse iteration require either $\mathrm{O}(n^3)$ or $\mathrm{O}(nk^2)$ operations in the worst case. Recent implementations of the Divide and Conquer method compete closely with our algorithm for matrices which allow extensive "deflations" but in general, they are slower and show an $\mathrm{O}(n^3)$ behavior to compute all the eigenpairs [3,10,15]. Our approach lends itself naturally to the computation of a small subset of the eigenpairs whereas the adaptation of Divide and Conquer to this case is somewhat artificial.

This paper is focused on presenting Algorithm MR^3 and proving that it computes eigenvectors that are numerically orthogonal and have small residual norms under prescribed conditions. In an earlier communication [6] we presented Algorithm Getvec which computes a very accurate approximation (error angle $= \mathrm{O}(n\varepsilon)$) to an eigenvector $\boldsymbol{v}$ of a tridiagonal matrix LDL^{t} (L is unit lower bidiagonal, D is diagonal) under two conditions: (a) the eigenvalue λ under consideration should have a large relative gap ($\geqslant$ tol say), and (b) the eigenpair $(\lambda, \boldsymbol{v})$ should be determined to high relative accuracy by L and D. More details are given in Section 2.1.

This paper considers the general scenario in which condition (a) need not be satisfied; in this case, Algorithm Getvec cannot guarantee numerically orthogonal eigenvectors. Failure of condition (a) occurs because the given matrix LDL^{t} has a cluster of eigenvalues $\lambda_l \leqslant \cdots \leqslant \lambda_r$ whose relative separation falls below an acceptable tolerance. To compute the corresponding eigenvectors, the proposed Algorithm MR^3 proceeds as follows. Choose a value τ close to the cluster (usually close to λ_l or λ_r) so that for at least one eigenvalue λ_k in the cluster, the gap relative to τ,

$$\mathrm{relgap}(\lambda_k - \tau) = \min_{l \leqslant i \leqslant r, i \neq k} \frac{|(\lambda_k - \tau) - (\lambda_i - \tau)|}{|\lambda_k - \tau|}$$

exceeds the acceptable tolerance. Then compute the new representation $L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}} = LDL^{\mathrm{t}} - \tau I$, where the subscript c stands for child. In exact arithmetic all eigenvectors $\boldsymbol{v}$ of LDL^{t} are eigenvectors of $L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}}$ with shifted eigenvalues $\lambda - \tau$. However this is not true for the computed $L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}}$ and the eigenvalues $\lambda - \tau$ need to be refined so that they have high relative accuracy with respect to the computed representation. Now, for all refined eigenvalues with relative gaps that exceed the tolerance, Algorithm MR^3 computes the corresponding eigenvectors by invoking Getvec using $L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}}$ and the refined eigenvalues. For the remaining eigenvalues, whose relative gaps still fall below the tolerance, the above procedure can be repeated. A detailed description of the algorithm is given in Section 2.2. Provided that the relevant eigenpairs $(\lambda - \tau, \boldsymbol{v})$ are determined to high relative accuracy by L_{c} and D_{c} then our algorithm will produce eigenvector approximations that are numerically orthogonal and have small residual norms. Proving these claims is a major part of this paper.

The preceding remarks suggest that Algorithm MR^3 computes a new representation of the tridiagonal matrix for each cluster of close eigenvalues. The algorithm computes a subset of eigenvector approximations from each representation and the reader may wonder whether the vectors computed from different representations will be numerically orthogonal. That is the first technical question addressed in this paper. If we begin with an initial form $L_0D_0L_0^{\mathrm{t}}$ (positive definite, say) and if every representation $L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}} = L_0D_0L_0^{\mathrm{t}} - \tau I$ were computed exactly then there would be nothing to prove since eigenvectors are invariant under translation.

It is the inevitable roundoff error in the computed factors that provokes this paper and a fundamental ingredient in the robustness of our approach is that we use special methods, called differential qd algorithms, to implement the transformation

$$L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}} = LDL^{\mathrm{t}} - \tau I$$

without explicitly forming the product LDL^{t}. These algorithms guarantee that the computed L_{c} and D_{c} are related to L and D by very special relative perturbations. Our proof of orthogonality hinges on the special nature of these perturbations and is somewhat complicated due to the many representations that may be used. It would be nice if a simple backward error analysis would let us say that each computed eigenvector is very close to an exact eigenvector of the initial matrix. That approach is doomed because some eigenvectors of the initial matrix (corresponding to very close eigenvalues perhaps even equal to working accuracy) may be poorly determined by the initial representation $L_0D_0L_0^{\mathrm{t}}$. Thus our preoccupation with high relative accuracy at intermediate stages in the algorithm is not for its own sake but to guarantee orthogonality.

Our second technical task is to show that each computed eigenpair $(\hat{\lambda}, z)$ has a small residual norm with respect to the initial matrix, $\|(L_0D_0L_0^{\mathrm{t}} - \hat{\lambda}I)z\| = \mathrm{O}(\varepsilon\|L_0D_0L_0^{\mathrm{t}}\|)$ where ε is the machine precision. In contrast to Getvec which delivers very accurate eigenvectors (error angle $= \mathrm{O}(n\varepsilon)$) for certain representations LDL^{t}, all we can ask, in general, when $L_0D_0L_0^{\mathrm{t}}$ has clusters of close eigenvalues, is that our computed vectors z have small residuals.

Our results are that the three properties given in Section 3.1 permit us to guarantee that the computed eigenpairs $(\hat{\lambda}_k, z_k)$ satisfy

$$\max_{j\neq k} |z_k^{\mathrm{t}} z_j| \leqslant 2[\mathscr{G} + (\mathrm{ndepth} - 1)\mathscr{R}]n\varepsilon \quad \text{(Theorem 3)},$$

and

$$\|L_0D_0L_0^{\mathrm{t}} z_k - z_k\hat{\lambda}_k\| \leqslant \mathscr{G}n\varepsilon + 9(\mathrm{ndepth} - 1)(2C + 1/2)\varepsilon\, \mathrm{spdiam}[L_0D_0L_0^{\mathrm{t}}] + \mathrm{O}(n\varepsilon^2) \quad \text{(Theorem 4)},$$

where $\mathscr{G}$ and $\mathscr{R}$ are specified by (3) and (16) respectively, C is a small constant often less than 1, ndepth is the depth of the representation tree (see Section 3) and spdiam denotes the spectral diameter, i.e., $\mathrm{spdiam}[A] = \lambda_{\max}[A] - \lambda_{\min}[A]$. Note that $\mathrm{spdiam}[A] \leqslant 2\|A\|$ for any symmetric A; in our algorithm the initial representation $L_0D_0L_0^{\mathrm{t}}$ is either indefinite or barely definite so that the situation $\|L_0D_0L_0^{\mathrm{t}}\| \gg \mathrm{spdiam}[L_0D_0L_0^{\mathrm{t}}]$ is avoided.

We now give a brief outline of the paper. Section 2 presents the proposed Algorithm MR^3 after reviewing Algorithm Getvec. To obtain both the orthogonality and residual results we use the idea of a representation tree that captures how Algorithm MR^3 acts on a given matrix. A certain amount of preparation is needed to introduce the representation tree which is provided in Section 3, and some example representation trees are illustrated in Section 3.2. Section 4 introduces the all important commutative diagram that captures the mixed relative roundoff error analysis of the differential qd transforms. Based on the commutative diagram, the orthogonality result is proved in Section 5 while Section 6 establishes the residual bound.

A word about notation. Throughout the paper we use ε to denote the roundoff unit but employ variations such as ε_i, η_j, η, ξ for small quantities that are not necessarily tied to a computer. T is used to denote a symmetric tridiagonal matrix, while we use LDL^{t} to denote its bidiagonal factorization. $(\lambda, \boldsymbol{v})$ denotes an exact eigenpair while $(\hat{\lambda}, \boldsymbol{z})$ denotes the computed approximation to the eigenpair. All angles θ between vectors/subspaces will be taken to be acute, so that we may write $\sin\theta$ instead of $|\sin\theta|$.

We assume that all tridiagonals are irreducible. If the given tridiagonal T is a direct sum of smaller matrices, or is close enough to such a matrix, then each block may be treated separately with considerable reduction in effort. So there is no loss in assuming that the off-diagonal entries exceed some threshold and so all eigenvalues are simple, even though they may be very close.

2. Algorithms

2.1. *Algorithm* Getvec

We first review Algorithm Getvec that was presented in [6] to compute an eigenvector of an isolated eigenvalue. Fig. 1 gives an outline of the algorithm, which takes an LDL^{t} factorization and an approximate eigenvalue $\hat{\lambda}$ as input and computes the corresponding eigenvector by forming the appropriate twisted factorization $N\varDelta N^{\mathrm{t}} = LDL^{\mathrm{t}} - \hat{\lambda}I$.

In [6] it was shown that the vector $\boldsymbol{z}$ computed by Getvec is an accurate approximation to a true eigenvector $\boldsymbol{v}$ of LDL^{t} provided the following two conditions hold:

Condition A. The eigenvalue λ should have an adequate relative separation from the rest of the spectrum, i.e.,

$$\mathrm{relgap}(\lambda) := \min_{\mu \neq \lambda} |\lambda - \mu|/|\lambda| \geqslant \mathrm{tol},$$

where μ ranges over all the other eigenvalues of LDL^{t}. Often tol is set to 10^{-3} which reflects a willingness to forego 3 digits of accuracy, see also [16, p. 322].

Algorithm Getvec(L,D, $\hat{\lambda}$)

Input: L is unit lower bidiagonal (l_i equals $L(i+1,i)$, $1 \leq i \leq n-1$), and D is diagonal (d_i equals $D(i,i)$, $1 \leq i \leq n$); LDL^t is the input tridiagonal matrix assumed to be irreducible. $\hat{\lambda}$ is an approximate eigenvalue (with small relative error).

Output: $\boldsymbol{z}$ is the computed eigenvector.

I. Factor $LDL^t - \hat{\lambda}I = L_+D_+L_+^t$ by the dstqds (differential stationary qd with shift) transform.

II. Factor $LDL^t - \hat{\lambda}I = U_-D_-U_-^t$ by the dqds (differential progressive qd with shift) transform.

III. Compute γ_k for $k = 1, \ldots, n$ by the formula $\gamma_k = s_k + \frac{d_k}{D_-(k+1)} p_{k+1}$ that involves the intermediate quantities s_k and p_{k+1} computed in the dstqds and dqds transforms (for details see [6, Sec 4.1]). In exact arithmetic, $\gamma_k^{-1} = [(LDL^t - \hat{\lambda})^{-1}]_{kk}$. Pick an r such that $|\gamma_r| = \min_k |\gamma_k|$. Form the twisted factors with twist index r, N and Δ, which satisfy $N\Delta N^t = LDL^t - \hat{\lambda}I$.

IV. Form the approximate eigenvector $\boldsymbol{z}$ by solving $N^t\boldsymbol{z} = \boldsymbol{e}_r$ ($\boldsymbol{e}_r$ is the r-th column of the identity matrix I) which is equivalent to solving $(LDL^t - \hat{\lambda}I)\boldsymbol{z} = N\Delta N^t\boldsymbol{z} = \boldsymbol{e}_r\gamma_r$ since $N\boldsymbol{e}_r = \boldsymbol{e}_r$ and $\Delta\boldsymbol{e}_r = \gamma_r\boldsymbol{e}_r$:

$$z(r) = 1.$$

$$\text{For } i = r-1, \ldots, 1, \quad z(i) = \begin{cases} -L_+(i)z(i+1), & z(i+1) \neq 0, \\ -(d_{i+1}l_{i+1}/d_il_i)z(i+2), & \text{otherwise.} \end{cases}$$

$$\text{For } j = r, \ldots, n-1, \quad z(j+1) = \begin{cases} -U_-(j)z(j), & z(j) \neq 0, \\ -(d_{j-1}l_{j-1}/d_jl_j)z(j-1), & \text{otherwise.} \end{cases}$$

Note that d_il_i is the $(i, i+1)$ element of LDL^t.

V. Compute $znrm = \|\boldsymbol{z}\|$ and set $\boldsymbol{z} \leftarrow \boldsymbol{z}/znrm$.

Fig. 1. Algorithm Getvec for computing the eigenvector of an isolated eigenvalue. See [6] for more details.

Condition B. The eigenpair $(\lambda, \boldsymbol{v})$ must be defined to high relative accuracy by L and D, i.e., small relative changes, $l_i \to l_i(1+\eta_i)$, $d_i \to d_i(1+\delta_i)$, $|\eta_i| < \xi$, $|\delta_i| < \xi$, $\xi \ll 1$, cause changes $\delta\lambda$ and $\delta\boldsymbol{v}$ that satisfy

$$|\delta\lambda| \leqslant K_1 n\xi|\lambda|, \quad \lambda \neq 0, \tag{1}$$

$$\sin\angle(\boldsymbol{v}, \boldsymbol{v} + \delta\boldsymbol{v}) \leqslant \frac{K_2 n\xi}{\text{relgap}(\lambda)} \tag{2}$$

for modest constants K_1 and K_2, say, smaller than 10. We call such an LDL^{t} factorization a relatively robust representation (RRR) for $(\lambda, \boldsymbol{v})$. Note that the case $\lambda = 0$ is easy since relative perturbations preserve the zero diagonal entry of D.

Details on twisted factorizations, differential qd transforms and Algorithm Getvec may be found in [6,12]. The following theorem quantifies the accuracy of the computed eigenvector:

Theorem 1 [6, Theorem 15]. *Let $(\lambda, \boldsymbol{v})$ be an eigenpair of an $n \times n$ real symmetric irreducible tridiagonal matrix LDL^{t} with $\|\boldsymbol{v}\| = 1$. Let $\hat{\lambda}$ be an accurate* *approximation closer to* λ *than to any other eigenvalue of* LDL^{t} *and let* $\boldsymbol{z}$ *be the vector computed in Step* IV *of Algorithm* Getvec. *Let* $\bar{L}$ *and* $\bar{D}$ *be perturbations of* L *and* D *determined by the error analysis of the differential twisted qd algorithm* (*see* [6, Section 5]) *and let* $(\bar{\lambda}, \bar{\boldsymbol{v}})$ *be the eigenpair of* $\bar{L}\bar{D}\bar{L}^{\mathrm{t}}$ *with* $\bar{\lambda}$ *the closest eigenvalue to* $\hat{\lambda}$ *and* $\|\bar{\boldsymbol{v}}\| = 1$. *Let* ε *denote the roundoff unit. Then*

$$\sin\angle(\boldsymbol{z}, \boldsymbol{v}) \leqslant \frac{(1+\varepsilon)^{5(n-1)} - 1}{(1-\varepsilon)^{5(n-1)}} + \frac{|\bar{\lambda} - \hat{\lambda}|}{|\bar{v}(r)|\operatorname{gap}(\hat{\lambda})}$$

$$+ \frac{(1+\varepsilon)^{6n-1} - 1}{2 - (1+\varepsilon)^{6n-1}} \cdot \operatorname{relcond}(\boldsymbol{v})$$

where $\operatorname{gap}(\hat{\lambda}) := \min\{|\hat{\lambda} - \bar{\mu}|, \bar{\mu} \in$ *spectrum of* $\bar{L}\bar{D}\bar{L}^{\mathrm{t}}, \bar{\mu} \neq \bar{\lambda}\}$, r *is the twist index in Step* III *of* Getvec *and* $\operatorname{relcond}(\boldsymbol{v})$ *is the relative condition number of* $\boldsymbol{v}$ *defined in* [6].

Let us consider the middle term in Theorem 1's bound. The choice of index r in Getvec is intended to pick out one of the largest entries in $\boldsymbol{v}$ while (1) ensures that λ can be computed to high relative accuracy. However in the proof of Theorem 15 in [6], $|\bar{\lambda} - \hat{\lambda}|/|\bar{v}(r)|$ appears as an upper bound for the quantity $|\bar{\gamma}_r|/\|\bar{z}\|$, which can be estimated from Algorithm Getvec. Thus if $|\bar{\gamma}_r|/\|\bar{z}\| \leqslant Kn\varepsilon|\hat{\lambda}|$ for some modest constant K, then under Condition A the middle term is bounded by $Kn\varepsilon/\mathrm{tol}$. Further, if $\operatorname{relcond}(\boldsymbol{v}) \leqslant \bar{K}$ then Theorem 1 guarantees that the vector $\boldsymbol{z}$ computed by Getvec satisfies

$$\sin\angle(\boldsymbol{z}, \boldsymbol{v}) \leqslant \mathscr{G}n\varepsilon \tag{3}$$

with $\mathscr{G}$ incorporating the details of Theorem 1; $\mathscr{G}$ will be a little larger than $5 + K/\mathrm{tol} + 6\bar{K}$. The dependence on tol can be removed since a good estimate of $\operatorname{gap}(\hat{\lambda})$ is easily available, and $|\bar{\gamma}_r|/(\|\bar{z}\|\operatorname{gap}(\bar{\lambda}))$ can often be driven to be smaller than $Kn\varepsilon$.

2.2. *Algorithm* MR^3

Fig. 2 describes Algorithm MR^3 that computes k eigenvectors of a symmetric tridiagonal matrix T in $\mathrm{O}(nk)$ time. The eigenvectors to be computed are specified by the index set Γ_0. The algorithm "breaks" each cluster of eigenvalues by shifting close to the cluster and forming a new factorization for each cluster. Note that no Gram–Schmidt orthogonalization is performed to obtain orthogonality.

Remark 1. The actual computation of each eigenvector is performed by invoking Algorithm Getvec when the (possibly shifted) eigenvalue has a large relative separation from its neighbors.

```
Algorithm MR$^3$($T$,$\Gamma_0$,$tol$)
Input: $T$ is the given symmetric tridiagonal (assumed to be irreducible),
       $\Gamma_0$ is the index set of desired eigenpairs,
       $tol$ is the input tolerance for relative gaps, usually $tol$ is set to $10^{-3}$.
Output: $(\hat\lambda_j, z_j), j \in \Gamma_0$, are the computed eigenpairs.

1. Choose $\mu$ such that $L_0D_0L_0^t = T + \mu I$ is a factorization that determines the desired eigenvalues
   and eigenvectors, $\lambda_j$ and $v_j$, $j \in \Gamma_0$, to high relative accuracy. In general, the shift $\mu$ can be
   in the interior of $T$'s spectrum, but a safe choice is to make $T + \mu I$ nearly positive or negative
   semi-definite so that we avoid the case of $||L_0D_0L_0^t|| \gg \mathrm{spdiam}(L_0D_0L_0^t)$.

2. Compute the desired eigenvalues of $L_0D_0L_0^t$ accurately enough to distinguish clusters (this can
   be done by the dqds algorithm [8] or by bisection using a differential qd transform). Form a
   work queue $Q$, and initialize $Q = \{(L_0, D_0, \Gamma_0)\}$.
   while ($Q$ is not empty) do
     3. Remove $(L, D, \Gamma)$ from the queue $Q$. Partition the computed eigenvalues $\hat\lambda_j, j \in \Gamma$, into
        clusters $\Gamma_1, \ldots, \Gamma_h$ according to their relative gaps and the input tolerance $tol$. The
        eigenvalues are thus designated as isolated (cluster size equals 1) or clustered. More
        precisely, if $\mathrm{rgap}(\hat\lambda_j) := \min_{i \neq j} |\hat\lambda_j - \hat\lambda_i|/|\hat\lambda_j| \geq tol$ then $\hat\lambda_j$ is isolated. On the other
        hand, all consecutive eigenvalues $\hat\lambda_{j-1}, \hat\lambda_j$ in a non-trivial cluster $\Gamma_c$ ($|\Gamma_c| > 1$) satisfy
        $|\hat\lambda_j - \hat\lambda_{j-1}|/|\hat\lambda_j| < tol$.
     4. For each cluster $\Gamma_c$, $c = 1, \ldots, h$, perform the following steps.
          If $|\Gamma_c| = 1$ with eigenvalue $\hat\lambda_j$, i.e., $\Gamma_c = \{j\}$, then
            a. If necessary, refine $\hat\lambda_j$ to have high relative accuracy with respect to $L$, $D$. Invoke
               Algorithm Getvec($L$,$D$,$\hat\lambda_j$) to obtain the computed eigenvector $z_j$.
          else
            b. Pick $\tau_c$ near the cluster and compute $LDL^t - \tau_c I = L_cD_cL_c^t$ using the dstqds (dif-
               ferential form of stationary qd) transform, see [6, Sec 4.1] for details. $(L_cD_cL_c^t, \Gamma_c)$
               will be called a child of $(LDL^t, \Gamma)$. Check for Property I (see Section 3.1), and if
               necessary, change $\tau_c$ and repeat.
            c. "Refine" the eigenvalues $\hat\lambda - \tau_c$ in $\Gamma_c$ so they have relative accuracy (with respect
               to the computed $L_c$, $D_c$) that is enough to distinguish between clusters.
               Set $\hat\lambda \leftarrow (\hat\lambda - \tau_c)_{refined}$, for all eigenvalues in $\Gamma_c$.
            d. Add $(L_c, D_c, \Gamma_c)$ to the queue $Q$.
          end if
       end for
   end while
```

Fig. 2. Algorithm MR^3 computes orthogonal eigenvectors by using multiple representations. Note that no Gram–Schmidt orthogonalization is needed.

Remark 2. In general, several intermediate factorizations LDL^t are needed. Each is called a "representation" and each is associated with a cluster of eigenvalues indexed by Γ. Each pair (LDL^t, Γ) is a node in a representation tree, which is defined in the next section.

Remark 3. For technical reasons we assume that Γ_0 is such that if Γ_0 indexes any eigenvalue in a cluster of $L_0D_0L_0^t$ then it contains *all* eigenvalues in that cluster. We make this assumption to simplify the algorithm and proofs which can be extended to handle an arbitrary Γ_0.

Remark 4. The crucial feature of Algorithm MR^3 concerns the representations $L_cD_cL_c^t \approx LDL^t - \tau_c I$ computed in Step 4b of Fig. 2. Each $L_cD_cL_c^t$ is associated uniquely with an index set Γ_c. Except for Γ_0 and singletons, each Γ_c corresponds to a cluster of eigenvalues—every eigenvalue of LDL^t in Γ_c has a relative gap that is smaller than the input threshold tol, usually set to 10^{-3}. The representations must satisfy three crucial properties: we defer describing these properties till Section 3.1 after introducing the notion of a representation tree in the next section.

3. The representation tree

As seen in the previous section, Algorithm MR^3 may use different representations to compute different eigenvectors. The sequence of computations performed by Algorithm MR^3 on a given matrix T can be neatly summarized by a representation tree, which was originally introduced in [5]. As we will see, representation trees facilitate our reasoning about the accuracy of the computed eigenvectors.

Before we introduce representation trees, we must resolve the practical question of how to designate eigenvalues. Because several different shifts will be employed we often denote each eigenvalue by its *index*, not its value. For example, eigenvalue $\{6\}$ always denotes the sixth eigenvalue from the left in the spectrum but its value will vary according to the current origin.

We assume that the reader is familiar with elementary concepts concerning graphs, particularly those related to a tree [2, Section 5.5]. A *representation tree* is a rooted tree where the root node denoted by $(\{L_0, D_0\}, \Gamma_0)$ or $(L_0D_0L_0^t, \Gamma_0)$ represents the initial RRR and initial index set Γ_0; the latter equals $\{1, 2, \ldots, n\}$ if the entire spectrum is desired. An example representation tree is shown in Fig. 3. Nodes

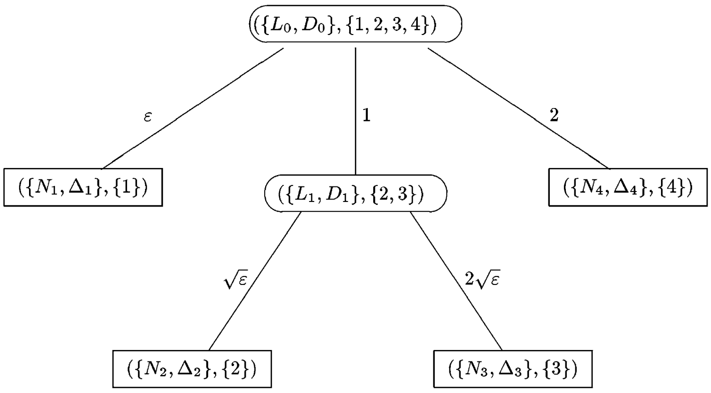
$(\{L_0, D_0\}, \{1,2,3,4\})$
ε
1
2
$(\{N_1, \Delta_1\}, \{1\})$
$(\{L_1, D_1\}, \{2,3\})$
$(\{N_4, \Delta_4\}, \{4\})$
$\sqrt{\varepsilon}$
$2\sqrt{\varepsilon}$
$(\{N_2, \Delta_2\}, \{2\})$
$(\{N_3, \Delta_3\}, \{3\})$

Fig. 3. An example representation tree.

that have no children are called *leaf nodes* while all other nodes are *internal*. The *depth* of a tree is the maximum number of edges on a path from a leaf node to the root. In general, an internal node in the tree is denoted by $(\{L_c, D_c\}, \varGamma_c)$ or $(L_cD_cL_c^t, \varGamma_c)$ where $\varGamma_c$ is a subset of the initial index set $\varGamma_0$. This node captures the fact that $L_cD_cL_c^t$ is an intermediate RRR used in the process of obtaining the eigenvectors indexed by $\varGamma_c$. The index set associated with any leaf node must be a singleton. A leaf node is denoted by $(\{N_i, \varDelta_i\}, \{i\})$ and captures the fact that the ith eigenvector is computed by Algorithm Getvec using the twisted factorization $N_i\varDelta_iN_i^t$ (see Section 2.1). An internal node $(LDL^t, \varGamma)$ may have h children (immediate descendants), some of them being internal nodes themselves and some being leaf nodes, with disjoint index sets $\varGamma_{\alpha_1}, \varGamma_{\alpha_2}, \ldots, \varGamma_{\alpha_h}$ that form a partition of $\varGamma$, i.e.,

$$\varGamma = \varGamma_{\alpha_1} \cup \varGamma_{\alpha_2} \cup \cdots \cup \varGamma_{\alpha_h}. \tag{4}$$

Note that this partition is determined by Algorithm MR^3 and depends on the relative separations between the eigenvalues of LDL^t. Algorithm MR^3 ensures that each internal node corresponds to a cluster, thus within each index set $\varGamma_{\alpha_i}$ the relative gaps with respect to LDL^t are below the input threshold tol, while across different index sets the relative gaps are larger than this tolerance. We elaborate on this property in Section 3.1.

Each edge connecting a parent node $(LDL^t, \varGamma)$ to its child node, $(L_cD_cL_c^t, \varGamma_c)$ or $(N_c\varDelta_cN_c^t, \varGamma_c)$, is labeled by a floating point number τ. Informally, this edge denotes the action of computing the new representation

$$LDL^t - \tau I = L_cD_cL_c^t \quad \text{or} \quad LDL^t - \tau I = N_c\varDelta_cN_c^t.$$

We discuss details of this computation in Section 4. We now present an example that leads to the representation tree of Fig. 3. Consider a 4 × 4 tridiagonal T_0 with eigenvalues

$$\lambda_1 = \varepsilon, \quad \lambda_2 = 1 + \sqrt{\varepsilon}, \quad \lambda_3 = 1 + 2\sqrt{\varepsilon}, \quad \lambda_4 = 2.0$$

with $\varepsilon \approx 2.2 \times 10^{-16}$ (machine precision in IEEE double precision arithmetic). Let $T_0 = L_0D_0L_0^t$ be the initial representation. Since relgap$(\lambda_1) \approx 1/\varepsilon$ and relgap$(\lambda_4) \approx 1$ are both large, Algorithm Getvec is invoked by MR^3 and the corresponding eigenvectors are computed using the twisted factorizations

$$L_0D_0L_0^t - \hat{\lambda}_1 I = N_1\varDelta_1N_1^t \quad \text{and} \quad L_0D_0L_0^t - \hat{\lambda}_4 I = N_4\varDelta_4N_4^t.$$

The interior eigenvalues have a relative separation of about $\sqrt{\varepsilon}$. Thus Algorithm MR^3 forms the new representation

$$L_0D_0L_0^t - I = L_1D_1L_1^t \quad (\text{taking } \tau = 1). \tag{5}$$

The relevant eigenvalues of $L_1D_1L_1^{\mathrm{t}}$ are now $\delta\lambda_2 \approx \lambda_2 - \tau = \sqrt{\varepsilon}$ and $\delta\lambda_3 \approx \lambda_2 - \tau = 2\sqrt{\varepsilon}$. Most importantly, these shifted eigenvalues now have large relative gaps. A vital feature of Algorithm MR^3 is that these eigenvalues are recomputed (refined is a more apt description) using the new matrices L_1 and D_1 till they have high relative accuracy with respect to $L_1D_1L_1^{\mathrm{t}}$. Finally, Algorithm Getvec is invoked to compute numerically orthogonal approximations z_2 and z_3 using the twisted factorizations

$$L_1D_1L_1^{\mathrm{t}} - \delta\lambda_2 I = N_2\Delta_2N_2^{\mathrm{t}} \quad \text{and} \quad L_1D_1L_1^{\mathrm{t}} - \delta\lambda_3 I = N_3\Delta_3N_3^{\mathrm{t}},$$

where $\delta\lambda_2 \approx \sqrt{\varepsilon}$ and $\delta\lambda_3 \approx 2\sqrt{\varepsilon}$.

The representation tree in Fig. 3 compactly summarizes the above computations. Many different matrices T_0 can produce the above behavior. Two contrasting examples, one with no element growth and the second with large element growth in forming $L_1D_1L_1^{\mathrm{t}}$ (see (5)), can be seen as Examples 5.1.1 and 5.1.2 in [5].

3.1. Properties

There are three properties that must be satisfied by the representations in the tree. The properties involve the choice of τ_{c} at each node of the tree and the “relative robustness” of the resulting computed representation $L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}$. The desired relationship is $LDL^{\mathrm{t}} - \tau_{\mathrm{c}}I = L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}$, however roundoff errors mean that the computed $L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}$ will be slightly different. The precise relationship between the computed and exact representations is given in Section 4.

Consider an internal node $(LDL^{\mathrm{t}}, \Gamma)$ in the tree and let $(L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}, \Gamma_{\mathrm{c}})$ be one of its child nodes. Some extra notation is needed to define the properties that each node representation must satisfy. We use $\mathscr{S}_\Gamma^A$ to denote the invariant subspace associated with Γ under the matrix A, i.e.,

$$A \cdot \mathscr{S}_\Gamma^A \subseteq \mathscr{S}_\Gamma^A.$$

Another way to describe $\mathscr{S}_\Gamma^A$ is as $\mathrm{span}(\boldsymbol{v}_i)$, $i \in \Gamma$, $A\boldsymbol{v}_i = \boldsymbol{v}_i\lambda_i$. We also define the relative gaps of the index set Γ with respect to the symmetric matrix A as

$$\mathrm{relgap}(\Gamma; A) := \min_{j \notin \Gamma, i \in \Gamma} |\lambda_i - \lambda_j|/|\lambda_i|, \quad \text{where } \lambda_i, \lambda_j \text{ are eigenvalues of } A.$$

We will use the subspaces $\mathscr{S}_{\Gamma_{\mathrm{c}}}^{L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}}$ and $\mathscr{S}_{\Gamma_{\mathrm{c}}}^{LDL^{\mathrm{t}}}$, and the relative gaps relgap $(\Gamma_{\mathrm{c}}; L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}})$ and $\mathrm{relgap}(\Gamma_{\mathrm{c}}; LDL^{\mathrm{t}})$ below. We are now ready to present the three properties that must be satisfied.

Property I (Relatively robust representation (RRR)). $L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}$ *should be an RRR for* Γ_{c}, *i.e., each eigenvalue* λ_i *of* $L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}$, $i \in \Gamma_{\mathrm{c}}$, *and the corresponding invariant subspace* $\mathscr{S}_{\Gamma_{\mathrm{c}}} = \mathscr{S}_{\Gamma_{\mathrm{c}}}^{L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}}$, *must be defined to high relative accuracy by* L_{c} *and* D_{c}.

This means that if $\bar{L}_{\rm c}\bar{D}_{\rm c}\bar{L}_{\rm c}^{\rm t}$ *is a small componentwise perturbation of* $L_{\rm c}D_{\rm c}L_{\rm c}^{\rm t}$, *i.e.*, $\bar{L}_{\rm c}(i, i+1) = L_{\rm c}(i, i+1)\cdot(1+\eta_i)$, $\bar{D}_{\rm c}(i,i) = D_{\rm c}(i,i)\cdot(1+\delta_i)$, $|\eta_i| < \xi$, $|\delta_i| < \xi$, $\xi \ll 1$, *then the perturbed quantities* $\bar{\lambda}_i$ *and* $\mathscr{S}_{\Gamma_{\rm c}}^{\bar{L}_{\rm c}\bar{D}_{\rm c}\bar{L}_{\rm c}^{\rm t}}$ *must satisfy*

$$|\bar{\lambda}_i - \lambda_i| \leqslant K_1 n\xi|\lambda_i|, \quad \lambda_i \neq 0, \tag{6}$$

$$\sin\angle\left(\mathscr{S}_{\Gamma_{\rm c}}^{L_{\rm c}D_{\rm c}L_{\rm c}^{\rm t}}, \mathscr{S}_{\Gamma_{\rm c}}^{\bar{L}_{\rm c}\bar{D}_{\rm c}\bar{L}_{\rm c}^{\rm t}}\right) \leqslant \frac{K_3 n\xi}{\mathrm{relgap}(\Gamma_{\rm c}; L_{\rm c}D_{\rm c}L_{\rm c}^{\rm t})}. \tag{7}$$

The angle between two subspaces is taken to be their largest principal angle, see [9, Section 12.4.3]. *Additionally, for the parent representation* $LDL^{\rm t}$ *and the same index set* $\Gamma_{\rm c}$,

$$\sin\angle\left(\mathscr{S}_{\Gamma_{\rm c}}^{LDL^{\rm t}}, \mathscr{S}_{\Gamma_{\rm c}}^{\bar{L}\bar{D}\bar{L}^{\rm t}}\right) \leqslant \frac{K_3 n\xi}{\mathrm{relgap}(\Gamma_{\rm c}; LDL^{\rm t})}, \tag{8}$$

where $\bar{L}\bar{D}\bar{L}^{\rm t}$ *is a small componentwise perturbation of* $LDL^{\rm t}$. *Note that* $\mathscr{S}_{\Gamma_{\rm c}}^{LDL^{\rm t}}$ *and* $\mathrm{relgap}(\Gamma_{\rm c}; LDL^{\rm t})$ *are quantities associated with the parent representation* $LDL^{\rm t}$, *but the child index set* $\Gamma_{\rm c}$. *The constants* K_1 *and* K_3 *are independent of the matrix and are of modest size* ($\leqslant$10 *say*).

Property II (Conditional element growth). *Let* $\mathrm{spdiam}[A]$ *denote the spectral diameter of a symmetric matrix* A, *i.e.*, $\mathrm{spdiam}[A] = \lambda_{\max}[A] - \lambda_{\min}[A]$. *For every representation* $LDL^{\rm t}$, *with index set* Γ, *and every computed eigenvector* z, $\|z\| = 1$, *with eigenvalue* λ *belonging to* Γ (*i.e.*, $\lambda = \lambda_j(LDL^{\rm t})$, $j \in \Gamma$) *we require*

$$\|Dz\| \leqslant C\cdot\mathrm{spdiam}[L_0D_0L_0^{\rm t}], \tag{9}$$

$$\|\overset{o}{L}\,D\,\overset{o}{L}{}^{\rm t} z\| \leqslant C\cdot\mathrm{spdiam}[L_0D_0L_0^{\rm t}], \quad \textit{where}\ \overset{o}{L} := L - I. \tag{10}$$

Notice that both D *and* $\overset{o}{L}\,D\,\overset{o}{L}{}^{\rm t}$ *are diagonal matrices and* $D + \overset{o}{L}\,D\,\overset{o}{L}{}^{\rm t} = \mathrm{diag}(LDL^{\rm t})$. *Here* C *is a modest constant, say* $C < 10$, *but often in practice* $C < 1$. *Note that this condition does allow large element growth in the factorization but only at indices that correspond to small entries in the computed vectors* z.

Property III (Relative gaps). *The eigenvalues of* $LDL^{\rm t}$ *in* Γ *are divided into subsets called children according to Step* 3 *of Algorithm* MR^3. *The purpose of this partition is to have relative gaps larger than* tol *between the children. To be specific, the shift* $\tau_{\rm c}$ *must be chosen so that*

$$\mathrm{relgap}(\Gamma_{\rm c}; LDL^{\rm t}) \geqslant \mathrm{tol}. \tag{11}$$

Moreover, the relative gap of the index set $\Gamma_{\rm c}$ *must increase on going down the tree, i.e., the relative gap with respect to the child representation*,

$$\mathrm{relgap}(\Gamma_{\rm c}; L_{\rm c}D_{\rm c}L_{\rm c}^{\rm t}) \geqslant \mathrm{relgap}(\Gamma_{\rm c}; LDL^{\rm t}) \geqslant \mathrm{tol}. \tag{12}$$

In (11) *and* (12) *the reference is to computed quantities and so* $L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t} = LDL^\mathrm{t} - \tau_\mathrm{c}I$ *may not hold. When it does then* (12) *follows from* (11) *as we now show. If* $L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t} = LDL^\mathrm{t} - \tau_\mathrm{c}I$ *then the eigenvalues of* $L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}$ *equal* $\lambda_i - \tau_\mathrm{c}$ *where* λ_i *is the* i*th eigenvalue of* LDL^t. *The shift* τ_c *is typically chosen very close to one end of the subset* $\{\lambda_i, i \in \Gamma_\mathrm{c}\}$ *of* LDL^t*'s spectrum in order that* $|\lambda_i - \tau_\mathrm{c}| \ll |\lambda_i|$, $i \in \Gamma_\mathrm{c}$. *For an arbitrary index pair* (i, j) *such that* $i \in \Gamma_\mathrm{c}$, $j \notin \Gamma_\mathrm{c}$ *we have*

$$\begin{aligned}|(\lambda_j - \tau_\mathrm{c}) - (\lambda_i - \tau_\mathrm{c})|/|\lambda_i - \tau_\mathrm{c}| &\geqslant \operatorname{relgap}(\Gamma_\mathrm{c}; LDL^\mathrm{t}) \cdot |\lambda_i|/|\lambda_i - \tau_\mathrm{c}| \\ &\gg \operatorname{relgap}(\Gamma_\mathrm{c}; LDL^\mathrm{t}).\end{aligned}$$

Then, *by* (11),

$$\operatorname{relgap}(\Gamma_\mathrm{c}; L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}) = \min_{i \in \Gamma_\mathrm{c}, j \notin \Gamma_\mathrm{c}} |(\lambda_j - \tau_\mathrm{c}) - (\lambda_i - \tau_\mathrm{c})|/|\lambda_i - \tau_\mathrm{c}| \geqslant \mathrm{tol}.$$

Thus the relevant relative gaps should increase as we go down the tree; *see Remark 5 in Section 4 for an example. Of course*, *Algorithm* MR^3 *does not know the exact eigenvalues of each representation*, *instead it uses estimates* $\hat{\lambda}_i$ *in order to approximate* (11) *and* (12). *By* (6), *it is possible to obtain estimates* $\hat{\lambda}_i$ *that have high relative accuracy.*

Although we have given Properties I and II separately, we suspect that they are very similar. Note that Property I is guaranteed for factorizations LDL^t that are definite [7], which is often our choice for the root representation $L_0D_0L_0^\mathrm{t}$, and also for LDL^t with L well conditioned for inversion [6].

Properties I and II are not as difficult to achieve as might appear at first sight. Except for Γ_0 every representation LDL^t need be an RRR for only the eigenvalues in Γ and these are the small eigenvalues $\lambda - \tau_\mathrm{c}$ (see Step 4b in Fig. 2). We conjecture that every almost singular factorization LDL^t is an RRR for the tiny eigenvalues but that is outside the scope of this paper, see [6,13]. Note that LDL^t always defines a zero eigenvalue to high relative accuracy.

Note also that we are free to choose the shifts τ_c to satisfy Properties I–III. A successful strategy for τ_c has been to try both the left and right extremes of Γ_c, choosing the end that leads to smaller element growth, i.e., minimizes $\|D_\mathrm{c}\|$. There is also an $\mathrm{O}(n)$ check for Property I, which can be used to validate the choice of τ_c. If both extremes fail we back off the cluster by the average gap. The triangular factorization $LDL^\mathrm{t} - \tau I$ is a rational function of τ and contains poles. We are only concerned with small neighborhoods just outside clusters of close eigenvalues and need only ensure that any shift τ_c not be too close to poles, if any, in these neighborhoods.

We will not say more about the choice of shifts or satisfaction of the three properties above since this material is beyond the scope of this paper; the reader is referred to [5,6,13] for more details. The purpose of this paper is to prove that Algorithm

MR^3 computes approximate eigenvectors that are numerically orthogonal and have small residual norms, assuming that the above properties hold.

3.2. Examples

We now present further examples to illustrate how a representation tree captures the action of Algorithm MR^3 on a given matrix. We urge the reader not to skip these examples.

Example 1 (*Nested clusters*). Consider a 13×13 real irreducible symmetric tridiagonal matrix T with spectrum ε, 1, 1 ± 10^{-15}, 1 ± 10^{-12}, 1 ± 10^{-9}, 1 ± 10^{-6}, 1 ± 10^{-3}, 2. Note that if it happens that the factorization $T - \mu I = L_0 D_0 L_0^{\mathrm{t}}$, with $\mu = 1 + \mathrm{O}(\varepsilon)$, is an RRR (Property I in Section 3.1 then it could be used as the initial representation instead of the positive definite factorization of T. The advantage would be that all the (shifted) eigenvalues would have large relative gaps; for example, for $\lambda = (1 + 10^{-6}) - \mu$, relgap$(\lambda)$ would approximately equal $\min\{10^{-6} - 10^{-9}, 10^{-3} - 10^{-6}\}/10^{-6} = 1 - 10^{-3}$. Hence all eigenvectors could be computed using this L_0 $D_0 L_0^{\mathrm{t}}$. However, we pass over this pleasant possibility in order to convey the way Algorithm MR^3 treats the general case.

For the initial representation $T = L_0 D_0 L_0^{\mathrm{t}}$ only the two extreme eigenvalues, ε and 2, have large relative gaps. The remaining eigenvalues are clustered around 1 and so it is reasonable to maintain that it is only the 11-dimensional invariant subspace associated with 1, 1 ± 10^{-15}, 1 ± 10^{-12}, 1 ± 10^{-9}, 1 ± 10^{-6}, 1 ± 10^{-3} that is well determined by L_0 and D_0, not the individual eigenvectors.

The first step in the algorithm is to invoke Algorithm Getvec, using L_0 and D_0, to compute eigenvectors for the extreme eigenvalues $\{1\}$ and $\{13\}$, with values ε and 2 respectively. We also locate a shift τ_1 at, or very little less than, the eigenvalue $1 - 10^{-3}$ so that $L_1 D_1 L_1^{\mathrm{t}} := L_0 D_0 L_0^{\mathrm{t}} - \tau_1 I$ yields an RRR for all the interior eigenvalues (other than ε and 2). These eigenvalues must be now refined so that they have high relative accuracy with respect to $L_1 D_1 L_1^{\mathrm{t}}$. The above computations are summarized by the partial representation tree in Fig. 4.

Now we describe the next levels of the tree produced by Algorithm MR^3. Eigenvalues $\{2\}$ and $\{12\}$ of $L_1 D_1 L_1^{\mathrm{t}}$ have the values 0 (or ε) and 2×10^{-3} and are

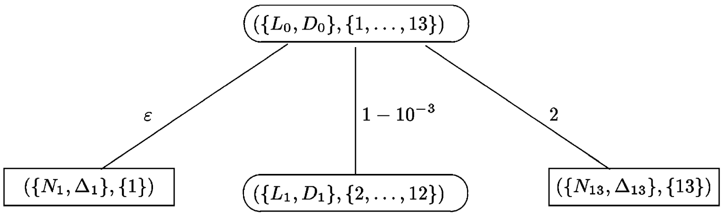

Fig. 4. Initial computations in the 13×13 example.

relatively well separated from $\{3, 4, \ldots, 11\}$. However, there are no large relative gaps in $\{3, 4, \ldots, 11\}$ although each eigenvalue is determined to high relative accuracy by L_1 and D_1. Thus Algorithm Getvec is invoked for $\{2\}$ and $\{12\}$ after these eigenvalues are refined using L_1 and D_1, and we get two leaves as children of $L_1 D_1 L_1^{\mathrm{t}}$ as shown in Fig. 5.

We must also find a shift τ_2 at, or very little less than, the eigenvalue $\{3\}$, with value $(1 - 10^{-6}) - (1 - 10^{-3})$, so that $L_2 D_2 L_2^{\mathrm{t}} := L_1 D_1 L_1^{\mathrm{t}} - \tau_2 I$ yields an RRR for the eigenvalue subset $\{3, 4, \ldots, 11\}$. It is not essential to locate τ_2 close to $\{3\}$. A value near $\{11\} = 10^{-3} + 10^{-6}$ that gives an RRR would be equally satisfactory.

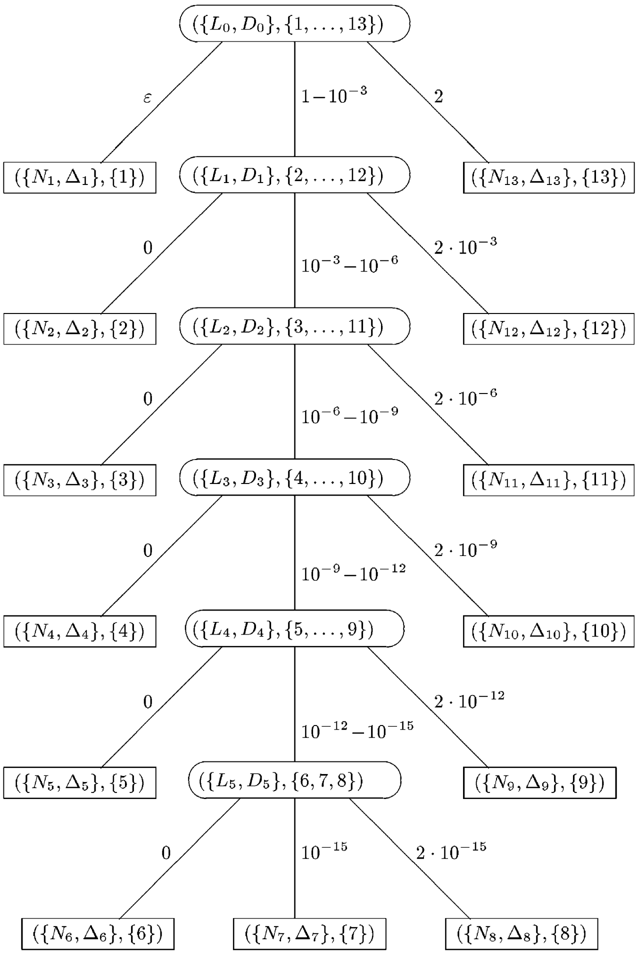
$(\{L_0, D_0\}, \{1, \ldots, 13\})$
ε
$1-10^{-3}$
2
$(\{N_1, \Delta_1\}, \{1\})$
$(\{L_1, D_1\}, \{2, \ldots, 12\})$
$(\{N_{13}, \Delta_{13}\}, \{13\})$
0
$10^{-3}-10^{-6}$
$2 \cdot 10^{-3}$
$(\{N_2, \Delta_2\}, \{2\})$
$(\{L_2, D_2\}, \{3, \ldots, 11\})$
$(\{N_{12}, \Delta_{12}\}, \{12\})$
0
$10^{-6}-10^{-9}$
$2 \cdot 10^{-6}$
$(\{N_3, \Delta_3\}, \{3\})$
$(\{L_3, D_3\}, \{4, \ldots, 10\})$
$(\{N_{11}, \Delta_{11}\}, \{11\})$
0
$10^{-9}-10^{-12}$
$2 \cdot 10^{-9}$
$(\{N_4, \Delta_4\}, \{4\})$
$(\{L_4, D_4\}, \{5, \ldots, 9\})$
$(\{N_{10}, \Delta_{10}\}, \{10\})$
0
$10^{-12}-10^{-15}$
$2 \cdot 10^{-12}$
$(\{N_5, \Delta_5\}, \{5\})$
$(\{L_5, D_5\}, \{6, 7, 8\})$
$(\{N_9, \Delta_9\}, \{9\})$
0
10^{-15}
$2 \cdot 10^{-15}$
$(\{N_6, \Delta_6\}, \{6\})$
$(\{N_7, \Delta_7\}, \{7\})$
$(\{N_8, \Delta_8\}, \{8\})$

Fig. 5. The entire representation tree for the extreme 13 × 13 example.

For simplicity we have assumed that shifts at the extreme left eigenvalue of each cluster yield RRRs. A shift within a cluster is also valid provided that an RRR is obtained but the danger of the shift failing is greater.

In this particular example, with shifts chosen from the left, each internal node has three children; two are leaves and the middle one is associated with a cluster. Finally we obtain the complete representation tree with depth 6 shown in Fig. 5. This is the maximum depth possible in IEEE double precision arithmetic when the tolerance for relative gaps, tol is set to $\min(10^{-3}, 1/n)$. Our analysis in Sections 5 and 6 will show why eigenvectors computed from different leaf nodes are numerically orthogonal and enjoy small residual norms with respect to the original representation $L_0D_0L_0^t$. For example, why are the computed eigenvectors z_3 and z_7 numerically orthogonal, why is $\|(L_0D_0L_0^t - \lambda_9 I)z_9\|$ small, etc.?

The above 13 × 13 example is extreme. To restore a sense of reality we mention a test matrix of order 966 that arose in the calculation of the energy levels in the biphenyl molecule, see [5, Section 6.4.1]. In this example, the root node had 805 children that were leaf nodes, and 63 children that were internal nodes or clusters. Of these 63 nodes, there were 49 with 2 eigenvalues each, 7 nodes with 3 eigenvalues each, 3 nodes with 4 eigenvalues each, 1 cluster of 5 eigenvalues, 2 clusters of 8 eigenvalues and 1 node with 9 eigenvalues. All nodes at the next level were leaf nodes, and the depth of the entire tree was 2.

Most matrices that we have encountered in our numerical experience have representation trees with depth 2 or 3. The 13 × 13 matrix of Example 1 has one cluster of eigenvalues with varying degrees of closeness, however all the eigenvalues are distinct. Although eigenvalues of irreducible tridiagonals are distinct they can be identical to working accuracy. The reader may wonder how Algorithm MR^3 can handle such a case.

Example 2 (*Eigenvalues identical to working accuracy*). Consider the 101 × 101 Wilkinson matrix W_{101}^{+} that has various eigenvalue clusters, each containing a pair of eigenvalues. The rightmost cluster is the tightest, with λ_{100} and λ_{101} identical to working accuracy. To 16 digits, the eigenvalues are

$$\lambda_{100} = 50.74619418290335, \quad \lambda_{101} = 50.74619418290335,$$

while $\lambda_{99} = 49.21067864733310$, and thus the rightmost cluster is well separated from the rest of the spectrum.

The seeming danger is that Algorithm MR^3 will not be able to “break” this tight cluster; the argument is that the eigenvalues are so close that shifting by a 16-digit floating point number τ will still lead to shifted eigenvalues that have all digits in common. Thus it seems that repeated shifting would be required to break a very tight pair.

However, this fear turns out to be unfounded. It is the inevitable roundoff errors that come to the rescue. In the case of W_{101}^{+}, if τ is chosen to equal $\mu +$

$\lambda_{100} = \mu + \lambda_{101}$ (where μ is the initial shift, $W_{101}^{+} + \mu I = L_0 D_0 L_0^{\mathrm{t}}$) then the differential qd transform is used to compute $L_0 D_0 L_0^{\mathrm{t}} - \tau I = L_1 D_1 L_1^{\mathrm{t}}$. There is roundoff error in computing L_1 and D_1 and due to this roundoff error, the smallest eigenvalues (in magnitude) of the computed $L_1 D_1 L_1^{\mathrm{t}}$, found after refinement, turn out to be

$$\delta\lambda_{100} = 6.4332181165285 \times 10^{-14} \quad \text{and} \quad \delta\lambda_{101} = 6.9956115177461 \times 10^{-14}.$$

The shifted eigenvalues $\delta\lambda_{100}$ and $\delta\lambda_{101}$ now have a large relative gap! The contamination in the eigenvalues, due to roundoff, prevents the shifted eigenvalues from sharing more digits. However, $\delta\lambda_{100}$ and $\delta\lambda_{101}$ have high relative accuracy with respect to the computed $L_1 D_1 L_1^{\mathrm{t}}$ and this allows Algorithm MR^3 to proceed and deliver numerically orthogonal eigenvectors. We have observed such behavior for many other matrices, for example, W_{501}^{+}, etc. Note also that in a software implementation, the value of τ can be varied if unfortunate symmetries in the data arise.

4. Commutative diagram

The representation tree involves various representations $L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}}$ that are formed by triangular factorization of $LDL^{\mathrm{t}} - \tau_{\mathrm{c}} I$. Accuracy in this calculation is crucial to the success of the algorithm. If L_{c} and D_{c} are computed in a straightforward manner by forming $LDL^{\mathrm{t}} - \tau_{\mathrm{c}} I$ and then factoring it, Algorithm MR^3 and Getvec would not always deliver accurate eigenvectors. The same verdict applies to Rutishauser's qd algorithm [14]. In [6] it was shown that if the differential form of the stationary qd algorithm (dstqds) is used, then special tiny relative changes in L, D and the computed $\hat{L}_{\mathrm{c}}$, $\hat{D}_{\mathrm{c}}$, yield ideal $\bar{L}$, $\bar{D}$, $\bar{L}_{\mathrm{c}}$, $\bar{D}_{\mathrm{c}}$ such that the relation

$$\bar{L}_{\mathrm{c}} \bar{D}_{\mathrm{c}} \bar{L}_{\mathrm{c}}^{\mathrm{t}} = \bar{L} \bar{D} \bar{L}^{\mathrm{t}} - \tau_{\mathrm{c}} I \tag{13}$$

holds exactly. This relationship is captured by the commutative diagram in Fig. 6.

Our goal is to show that vectors z_i and z_j computed by Algorithm MR^3 from different representations are orthogonal to working accuracy. The proof is somewhat complicated and brings in diverse aspects of the algorithm. Consequently we must break the argument into pieces and must introduce some notation.

Angle with parent subspace. Consider the commutative diagram in Fig. 6, which corresponds to the computation of a typical child node $(L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}}, \Gamma_{\mathrm{c}})$ of the parent $(LDL^{\mathrm{t}}, \Gamma)$. Assume that Γ_{c} is not a singleton. There are two subspaces associated with Γ_{c}; one is $\mathscr{S}_{\Gamma_{\mathrm{c}}}^{L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}}}$, the subspace invariant under $L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}}$, the other is $\mathscr{S}_{\Gamma_{\mathrm{c}}}^{LDL^{\mathrm{t}}}$, the subspace invariant under LDL^{t}. We define the angle

$$\Phi_{\Gamma_{\mathrm{c}}} := \angle\left(\mathscr{S}_{\Gamma_{\mathrm{c}}}^{L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}}}, \mathscr{S}_{\Gamma_{\mathrm{c}}}^{LDL^{\mathrm{t}}}\right), \quad 0 \leqslant \Phi_{\Gamma_{\mathrm{c}}} \leqslant \frac{\pi}{2}, \tag{14}$$

to be the largest principal angle between the two subspaces [9, Section 12.4.3]. In exact arithmetic, i.e., if there were no roundoff errors in computing $L_{\mathrm{c}} D_{\mathrm{c}} L_{\mathrm{c}}^{\mathrm{t}}$, by the

$$\begin{tikzcd}[column sep=huge, row sep=huge]
L, D \arrow[r, "\textbf{dstqds}", "\text{computed}"'] \arrow[d, "{\begin{array}{l}\text{change each}\\ D(i,i) \text{ by } 1\ \mathit{ulp},\\ L(i,i-1) \text{ by } 3\ \mathit{ulp}\text{s.}\end{array}}"'] & L_c, D_c \\
\bar{L}, \bar{D} \arrow[r, "\textbf{dstqds}", "\text{exact}"'] & \bar{L}_c, \bar{D}_c \arrow[u, "{\begin{array}{l}\text{change each}\\ D_c(i,i) \text{ by } 2\ \mathit{ulp}\text{s},\\ L_c(i,i-1) \text{ by } 3\ \mathit{ulp}\text{s.}\end{array}}"']
\end{tikzcd}$$

$$\bar{L}\bar{D}\bar{L}^t - \tau_c I = \bar{L}_c\bar{D}_c\bar{L}_c^t$$

Fig. 6. Commutative diagram (the acronym *ulp* stands for *u*nits in the *l*ast *p*lace held).

translation invariance of subspaces, $\mathscr{S}_{\Gamma_\mathrm{c}}^{L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}}$ and $\mathscr{S}_{\Gamma_\mathrm{c}}^{LDL^\mathrm{t}}$ would be *identical* and Φ_{Γ_c} would be zero. Reassuringly, as shown below in Lemma 1, thanks to the properties of differential qd algorithms the subspaces are close even in finite precision arithmetic.

Lemma 1. *Let (LDL^t, Γ) be a non-leaf node in the representation tree and let $(L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}, \Gamma_\mathrm{c})$ be one of its child nodes, where $L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}$ is the factorization of $LDL^\mathrm{t} - \tau_\mathrm{c}I$ computed by the* dstqds *transform. Then the commutative diagram in Fig.* 6 *holds and*

$$\sin\Phi_{\Gamma_\mathrm{c}} := \sin\angle\big(\mathscr{S}_{\Gamma_\mathrm{c}}^{L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}}, \mathscr{S}_{\Gamma_\mathrm{c}}^{LDL^\mathrm{t}}\big) \leqslant \mathscr{R}n\varepsilon,$$

where $\mathscr{R}$ is a constant that depends on perturbation properties of LDL^t and $L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}$, on the error analysis of the dstqds *transform and on the relative gap tolerance* tol *used in Step* 3 *of Algorithm* MR^3.

Proof. Since the dstqds transform is used to compute $L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}$, Theorem 2 in [6] shows that the commutative diagram of Fig. 6 holds; in particular the relation (13) holds. By translation invariance of invariant subspaces

$$\mathscr{S}_{\Gamma_\mathrm{c}}^{\bar{L}\bar{D}\bar{L}^\mathrm{t}} = \mathscr{S}_{\Gamma_\mathrm{c}}^{\bar{L}_\mathrm{c}\bar{D}_\mathrm{c}\bar{L}_\mathrm{c}^\mathrm{t}}$$

and so, taking all angles as acute,

$$\begin{aligned}\Phi_{\Gamma_\mathrm{c}} &= \angle\big(\mathscr{S}_{\Gamma_\mathrm{c}}^{L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}}, \mathscr{S}_{\Gamma_\mathrm{c}}^{LDL^\mathrm{t}}\big)\\ &\leqslant \angle\big(\mathscr{S}_{\Gamma_\mathrm{c}}^{LDL^\mathrm{t}}, \mathscr{S}_{\Gamma_\mathrm{c}}^{\bar{L}\bar{D}\bar{L}^\mathrm{t}}\big) + \angle\big(\mathscr{S}_{\Gamma_\mathrm{c}}^{L_\mathrm{c}D_\mathrm{c}L_\mathrm{c}^\mathrm{t}}, \mathscr{S}_{\Gamma_\mathrm{c}}^{\bar{L}_\mathrm{c}\bar{D}_\mathrm{c}\bar{L}_\mathrm{c}^\mathrm{t}}\big). \qquad (15)\end{aligned}$$

Informally the first term on the right in (15) reflects the sensitivity of the left side of the commutative diagram, while the second reflects the sensitivity of the right side. We next show that both these terms are small under our assumptions.

Looking at both sides of the commutative diagram we see that no entry changes by more than $(1+3\varepsilon)$. By assumption, $L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}$ is an RRR for Γ_{c}, so by (7) of Property I of an RRR,

$$\sin\angle\left(\mathscr{S}_{\Gamma_{\mathrm{c}}}^{L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}}, \mathscr{S}_{\Gamma_{\mathrm{c}}}^{\bar{L}_{\mathrm{c}}\bar{D}_{\mathrm{c}}\bar{L}_{\mathrm{c}}^{\mathrm{t}}}\right) \leqslant \frac{K_3 n(3\varepsilon)}{\operatorname{relgap}(\Gamma_{\mathrm{c}};\, L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}})} \leqslant \frac{K_3 n(3\varepsilon)}{\mathrm{tol}},$$

where the last inequality follows from (12). Similarly, by (8) and (11),

$$\sin\angle\left(\mathscr{S}_{\Gamma_{\mathrm{c}}}^{LDL^{\mathrm{t}}}, \mathscr{S}_{\Gamma_{\mathrm{c}}}^{\bar{L}\bar{D}\bar{L}^{\mathrm{t}}}\right) \leqslant \frac{K_3 n(3\varepsilon)}{\operatorname{relgap}(\Gamma_{\mathrm{c}};\, LDL^{\mathrm{t}})} \leqslant \frac{K_3 n(3\varepsilon)}{\mathrm{tol}}.$$

Insert these bounds in (15) to obtain

$$\sin\Phi_{\Gamma_{\mathrm{c}}} = \sin\angle\left(\mathscr{S}_{\Gamma_{\mathrm{c}}}^{L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}}, \mathscr{S}_{\Gamma_{\mathrm{c}}}^{LDL^{\mathrm{t}}}\right) \leqslant \mathscr{R} n\varepsilon \quad \text{with } \mathscr{R} := 6K_3/\mathrm{tol}. \tag{16}$$

□

Remark 5. The example in Fig. 5 shows that the relevant relative gaps can be far greater than tol, and illustrates the general phenomenon that relgap(Γ) tends to increase as Γ moves down the representation tree. For example,

$$\operatorname{relgap}(\{6,7,8\};\, L_3D_3L_3^{\mathrm{t}}) \approx \frac{(10^{-9}+10^{-12})-(10^{-9}+10^{-15})}{10^{-9}+10^{-15}} \approx 10^{-3},$$

$$\operatorname{relgap}(\{6,7,8\};\, L_4D_4L_4^{\mathrm{t}}) \approx \frac{2\times 10^{-12}-(10^{-12}+10^{-15})}{10^{-12}+10^{-15}} \approx 1,$$

while

$$\operatorname{relgap}(\{6,7,8\};\, L_5D_5L_5^{\mathrm{t}}) \approx \frac{(10^{-12}+10^{-15})-2\times 10^{-15}}{2\cdot 10^{-15}} = \frac{1}{2}(10^3-1).$$

The reader may find it beneficial to instantiate the proof of Lemma 1 for any non-leaf node of the representation tree in Fig. 5.

5. Orthogonality

Algorithm MR^3 begins with a symmetric tridiagonal matrix in factored form $L_0D_0L_0^{\mathrm{t}}$ that defines all the desired eigenvalues to high relative accuracy. The root node of the representation tree is the pair $(L_0D_0L_0^{\mathrm{t}}, \Gamma_0)$, Γ_0 being the initial index set. Leaf nodes are given by the pairs $(N_j\varDelta_jN_j^{\mathrm{t}}, \{j\})$, and signify the computation of the approximate eigenvector z_j by Algorithm Getvec using the twisted factorization $N_j\varDelta_jN_j^{\mathrm{t}}$. The computed vector z_j has the attractive property that it differs from the eigenvector of the parent LDL^{t} for eigenvalue $\{j\}$ by $\mathrm{O}(n\varepsilon)$, under prescribed conditions (see Theorem 1). To show accuracy of the computed eigenvectors we will relate them to representations up the tree; however, it is not possible to show that each computed vector is within $\mathrm{O}(n\varepsilon)$ of the corresponding eigenvector of the root $L_0D_0L_0^{\mathrm{t}}$ since individual eigenvectors of $L_0D_0L_0^{\mathrm{t}}$ may not be determined to such high accuracy. Instead, in this section we show why the vectors computed from different representations are orthogonal to working accuracy.

Note that each node of the representation tree has a distinct index set Γ, and thus we will often denote the node by its index set. The quantity of interest is

$$\mathrm{dot}_{\Gamma} = \max_{i\in\Gamma, j\in\Gamma} \cos\angle(z_i, z_j), \quad i \neq j, \tag{17}$$

where z_i and z_j are the computed eigenvectors. As our main result, in Theorem 3 we give a bound on dot_{Γ_0} where Γ_0 is the index set for the root node.

To present our main orthogonality result, we need some additional notation. Recall that eigenvectors are invariant under translation and consequently we denote eigenvalues simply by an index; $\{q\}$ denotes the qth eigenvalue from the left, $\lambda_q < \lambda_{q+1}$. As in Section 4, let $\mathscr{S}_{\Gamma}^{LDL^{\mathrm{t}}}$ denote the subspace invariant under LDL^{t} associated with Γ. For ease of use, we will often denote $\mathscr{S}_{\Gamma}^{LDL^{\mathrm{t}}}$ by $\mathscr{S}_{\Gamma}$ for the internal node $(LDL^{\mathrm{t}}, \Gamma)$. In addition there is a set of computed normalized vectors associated with Γ, i.e., $\{z_i; i \in \Gamma\}$. To understand how the quantity defined in (17) propagates up the tree we need to define the following additional angle associated with the subspace $\mathscr{S}_{\Gamma}$.

Angle with computed vectors. For any internal node Γ and any $k \in \Gamma$, define

$$\Psi_{k,\Gamma} := \angle\left(z_k, \mathscr{S}_{\Gamma}\right), \tag{18}$$

where, as usual, $\angle(z_k, \mathscr{S}_{\Gamma})$ denotes the smallest angle made by z_k with any vector in the subspace $\mathscr{S}_{\Gamma}$, i.e., the principal angle between $\mathscr{S}_{\Gamma}$ and the 1-dimensional subspace spanned by z_k. Trivially, $\Psi_{k,\Gamma_0} = 0$ when $\Gamma_0 = \{1, \ldots, n\}$.

At a leaf: As a special case consider the parent $(LDL^{\mathrm{t}}, \Gamma)$ of the leaf node $\{k\}$ at which z_k is computed by Algorithm Getvec. The main result of [6] is that there is an eigenvector v_k of LDL^{t} such that $\sin\angle(z_k, v_k) \leqslant \mathscr{G}n\varepsilon$, for a small constant $\mathscr{G}$ $(= \mathrm{O}(1))$ that incorporates the large relative gap of $\{k\}$ in the spectrum of LDL^{t}; see Theorem 1 and (3). Since $k \in \Gamma$, $v_k \in \mathscr{S}_{\Gamma}$ and thus

$$\sin\Psi_{k,\Gamma} \leqslant \sin\angle(z_k, v_k) \leqslant \mathscr{G}n\varepsilon. \tag{19}$$

Away from the leaf: For nodes $(LDL^{\mathrm{t}}, \Gamma)$ nearer the root, the index set Γ that includes k increases and $\Psi_{k,\Gamma}$ should (weakly) decrease. In fact, if there were no roundoff error in computing each new representation, the Ψ's would decrease monotonically as we go up the tree. However, in the presence of roundoff errors the situation is more complicated and the angle Φ_Γ defined in Section 4 comes into play, as shown in Lemma 2 below.

Consider an internal node $(LDL^{\mathrm{t}}, \Gamma)$. Let $\mathscr{S}_\Gamma^{\text{parent}}$ denote the subspace associated with the index set Γ that is invariant under the *parent matrix* of the node $(LDL^{\mathrm{t}}, \Gamma)$. We contrast $\Psi_{k,\Gamma}$ with Φ_Γ $(\equiv \angle(\mathscr{S}_\Gamma, \mathscr{S}_\Gamma^{\text{parent}}))$ defined in (18) and (14) respectively. The angle Φ_Γ is independent of Algorithm Getvec and depends on how well the parent and child representations determine their invariant subspaces $\mathscr{S}_\Gamma^{\text{parent}}$ and $\mathscr{S}_\Gamma$ $(\equiv \mathscr{S}_\Gamma^{LDL^{\mathrm{t}}})$ respectively. On the other hand, $\Psi_{k,\Gamma}$ depends on the eigenvector z_k computed at the leaf node by Algorithm Getvec. Note that we have defined $\Psi_{k,\Gamma}$ and Φ_Γ only for the internal nodes of the representation tree; there is no need to define these angles for leaf nodes.

Our assumption that each representation is an RRR for its index set Γ guarantees, by Lemma 1, that the Φ_Γ are small. The bound (19) which follows from [6] guarantees that $\Psi_{k,\Gamma}$ is small at the parent (Γ) of the leaf node $\{k\}$. The following lemma gives a recurrence for $\Psi_{k,\Gamma}$ that allows us to understand how this angle grows as we go up the tree to the root node.

Lemma 2. *We consider z_j computed by* Getvec. *Let Γ be an internal node in the representation tree with $j \in \Gamma$. Let Γ_α be the* (*unique*) *child node*, *i.e.*, *immediate descendant*, *of Γ such that $j \in \Gamma_\alpha$* (*as illustrated in Fig.* 7). *Then*

$$\sin \Psi_{j,\Gamma} \leqslant \sin \Psi_{j,\Gamma_\alpha} + \sin \Phi_{\Gamma_\alpha}. \tag{20}$$

Proof. Taking all angles as acute, for any subspace $\mathscr{U}$

$$\sin \Psi_{j,\Gamma} = \sin \angle(z_j, \mathscr{S}_\Gamma) \leqslant \sin \angle(z_j, \mathscr{U}) + \sin \angle(\mathscr{U}, \mathscr{S}_\Gamma).$$

It is convenient to take $\mathscr{U} = \mathscr{S}_{\Gamma_\alpha}$. Note that the subspaces $\mathscr{S}_{\Gamma_\alpha}$ and $\mathscr{S}_\Gamma$ are of different dimensions and $\angle(\mathscr{S}_{\Gamma_\alpha}, \mathscr{S}_\Gamma)$ is again the largest principal angle between the subspaces. Since Γ_α is an immediate descendant of Γ, $\Gamma_\alpha \subseteq \Gamma$ and $\mathscr{S}_{\Gamma_\alpha}^{\text{parent}} \subseteq \mathscr{S}_\Gamma$ $\big($in Fig. 7, by definition, $\mathscr{S}_{\Gamma_\alpha}^{\text{parent}}$ equals $\mathscr{S}_{\Gamma_\alpha}^{LDL^{\mathrm{t}}}$ while $\mathscr{S}_\Gamma$ equals $\mathscr{S}_\Gamma^{LDL^{\mathrm{t}}}\big)$. Thus

$$\sin \angle(\mathscr{S}_{\Gamma_\alpha}, \mathscr{S}_\Gamma) \leqslant \sin \angle\big(\mathscr{S}_{\Gamma_\alpha}, \mathscr{S}_{\Gamma_\alpha}^{\text{parent}}\big) = \sin \Phi_{\Gamma_\alpha}, \tag{21}$$

and the result (20) holds. □

Given two different leaf nodes $\{j\}$ and $\{k\}$ in the representation tree, they have a unique least common ancestor, which is the first internal node up the tree that is an ancestor of both the leaves. For example, in Fig. 5, the least common ancestor of $\{3\}$ and $\{7\}$ is $(L_2 D_2 L_2^{\mathrm{t}}, \{3, \ldots, 11\})$. In ascending the tree the level of orthogonality among the computed eigenvectors is governed by the following lemma.

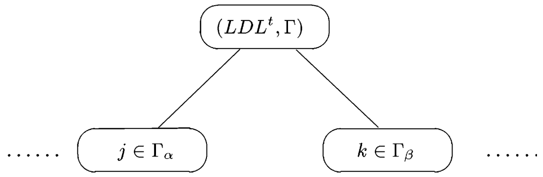

Fig. 7. Relation of Γ to Γ_α and Γ_β.

Lemma 3. *Consider the pair of computed eigenvectors* z_j *and* z_k, *and let* $(LDL^{\mathrm{t}}, \Gamma)$ *be their least common ancestor. Let* Γ_α *and* Γ_β *be the child nodes* (*immediate descendants*) *of* Γ *that contain the indices* j *and* k *respectively* $(\Gamma_\alpha \cap \Gamma_\beta = \phi)$. *Assuming that both* z_j *and* z_k *satisfy* (19), *then*

$$\cos\angle(z_j, z_k) \leqslant \begin{cases} \sin\Psi_{j,\Gamma_\alpha} + \sin\Phi_{\Gamma_\alpha} + \sin\Psi_{k,\Gamma_\beta} + \sin\Phi_{\Gamma_\beta} \\ \quad \textit{if } \Gamma_\alpha \textit{ and } \Gamma_\beta \textit{ are internal nodes}, \\ \mathscr{G}n\varepsilon + \sin\Psi_{k,\Gamma_\beta} + \sin\Phi_{\Gamma_\beta} \\ \quad \textit{if only } \Gamma_\alpha \textit{ is a leaf node}, \\ \mathscr{G}n\varepsilon + \sin\Psi_{j,\Gamma_\alpha} + \sin\Phi_{\Gamma_\alpha} \\ \quad \textit{if only } \Gamma_\beta \textit{ is a leaf node}, \\ 2\mathscr{G}n\varepsilon \\ \quad \textit{if } \Gamma_\alpha \textit{ and } \Gamma_\beta \textit{ are leaf nodes}. \end{cases}$$

Proof. Consider the case when both Γ_α and Γ_β are internal nodes with parent Γ as illustrated in Fig. 7. Recall the subspaces $\mathscr{S}_{\Gamma_\alpha}$, $\mathscr{S}_{\Gamma_\alpha}^{\text{parent}}$, $\mathscr{S}_{\Gamma_\beta}$ and $\mathscr{S}_{\Gamma_\beta}^{\text{parent}}$. Since Γ is the parent of Γ_α and Γ_β both $\mathscr{S}_{\Gamma_\alpha}^{\text{parent}}$ and $\mathscr{S}_{\Gamma_\beta}^{\text{parent}}$ are subspaces contained in $\mathscr{S}_\Gamma$. Since LDL^{t} is real and symmetric, and $\Gamma_\alpha \cap \Gamma_\beta = \phi$,

$$\mathscr{S}_{\Gamma_\alpha}^{\text{parent}} \perp \mathscr{S}_{\Gamma_\beta}^{\text{parent}}.$$

Hence, taking all angles as acute,

$$\begin{aligned} \cos\angle(z_j, z_k) &\leqslant \sin\angle\big(z_j, \mathscr{S}_{\Gamma_\alpha}^{\text{parent}}\big) + \sin\angle\big(z_k, \mathscr{S}_{\Gamma_\beta}^{\text{parent}}\big) \\ &\leqslant \sin\angle(z_j, \mathscr{S}_{\Gamma_\alpha}) + \sin\angle\big(\mathscr{S}_{\Gamma_\alpha}, \mathscr{S}_{\Gamma_\alpha}^{\text{parent}}\big) \\ &\quad + \sin\angle(z_k, \mathscr{S}_{\Gamma_\beta}) + \sin\angle\big(\mathscr{S}_{\Gamma_\beta}, \mathscr{S}_{\Gamma_\beta}^{\text{parent}}\big) \\ &= \sin\Psi_{j,\Gamma_\alpha} + \sin\Phi_{\Gamma_\alpha} + \sin\Psi_{k,\Gamma_\beta} + \sin\Phi_{\Gamma_\beta} \end{aligned} \tag{22}$$

by definitions (18) and (14).

We now consider the special case when one or both of the children are leaves. If Γ_α is a leaf, i.e., $\Gamma_\alpha = \{j\}$, we can directly use (19) to give

$$\sin\angle\big(z_j, \mathscr{S}_{\Gamma_\alpha}^{\text{parent}}\big) \leqslant \mathscr{G}n\varepsilon.$$

When both Γ_α and Γ_β are leaves then, by (22),

$$\cos\angle(z_j, z_k) \leqslant 2\mathscr{G}n\varepsilon,$$

while if only Γ_α is a leaf then

$$\cos\angle(z_j, z_k) \leqslant \mathscr{G}n\varepsilon + (\sin\Psi_{k,\Gamma_\beta} + \sin\Phi_{\Gamma_\beta}). \qquad \square$$

Theorem 2. *Consider the pair of computed eigenvectors z_j and z_k, and let $(LDL^{\rm t}, \Gamma)$ be their least common ancestor. Let* depth(Γ, j) *and* depth(Γ, k) *be the number of edges on the path from node Γ to the leaf nodes $\{j\}$ and $\{k\}$ respectively. Let the vectors z_j and z_k satisfy the bound in* (19) *(due to Algorithm* Getvec), *and let all internal nodes in the representation tree satisfy* (16). *Then*

$$\cos\angle(z_j, z_k) \leqslant 2\mathscr{G}n\varepsilon + \{\text{depth}(\Gamma, j) + \text{depth}(\Gamma, k) - 2\}\mathscr{R}n\varepsilon.$$

Proof. The proof follows by applying the recurrence for $\sin\Psi_{k,\Gamma}$ given in Lemma 2 to the bound for $\cos\angle(z_j, z_k)$ in Lemma 3. □

Finally, we get the desired bound on the worst dot product between the eigenvectors computed by Algorithm MR^3.

Theorem 3. *Let Γ be a non-leaf node and let* dot_Γ *be as defined in* (17). *Using the notation of Theorem* 2,

$$\text{dot}_\Gamma \leqslant 2\mathscr{G}n\varepsilon + \max_{j,k\in\Gamma}\{\text{depth}(\Gamma, j) + \text{depth}(\Gamma, k) - 2\}\mathscr{R}n\varepsilon.$$

Thus, for the root node Γ_0,

$$\text{dot}_{\Gamma_0} \leqslant 2\mathscr{G}n\varepsilon + 2(\text{ndepth} - 1)\mathscr{R}n\varepsilon,$$

where ndepth *is the depth of the representation tree.*

We now illustrate the above theorems on the 13 × 13 example of Fig. 5. Consider z_3 and z_7. Their least common ancestor is $(L_2D_2L_2^{\rm t}, \Gamma_2)$ with $\Gamma_2 = \{3, \ldots, 11\}$, depth$(\Gamma_2, 3) = 1$ and depth$(\Gamma_2, 7) = 4$. Thus, according to Theorem 2,

$$\cos\angle(z_3, z_7) \leqslant 2\mathscr{G}n\varepsilon + 3\mathscr{R}n\varepsilon.$$

Examining Fig. 5, it is instructive to follow the path taken by the proof of Theorem 2 for this pair:

$$\begin{aligned}
\cos\angle(z_3, z_7) &\leqslant \sin\angle\big(z_3, \mathscr{S}^{\text{parent}}_{\{3\}}\big) + \sin\angle\big(z_7, \mathscr{S}^{\text{parent}}_{\{4,\ldots,10\}}\big),\\
&\quad\text{by (22)},\\
&\leqslant \mathscr{G}n\varepsilon + \sin\angle\big(z_7, \mathscr{S}_{\{4,\ldots,10\}}\big) + \sin\angle\big(\mathscr{S}_{\{4,\ldots,10\}}, \mathscr{S}^{\text{parent}}_{\{4,\ldots,10\}}\big),\\
&\quad\text{by (19)},\\
&= \mathscr{G}n\varepsilon + \sin\Psi_{7,\{4,\ldots,10\}} + \sin\Phi_{\{4,\ldots,10\}},\\
&\quad\text{by (18) and (14)},\\
&\leqslant \mathscr{G}n\varepsilon + \sin\Psi_{7,\{5,\ldots,9\}} + \sin\Phi_{\{5,\ldots,9\}} + \mathscr{R}n\varepsilon,\\
&\quad\text{by (20) and (16)},\\
&\leqslant \mathscr{G}n\varepsilon + 2\mathscr{R}n\varepsilon + \sin\Psi_{7,\{6,7,8\}} + \sin\Phi_{\{6,7,8\}},\\
&\quad\text{by (20) and (16)},
\end{aligned}$$

$$\leqslant 2\mathscr{G}n\varepsilon + 3\mathscr{R}n\varepsilon,$$
by (19) and (16).

6. Residual norms

The remaining question to settle in this paper is: do the computed eigenvectors enjoy small residual norms relative to the initial representation $T_0 = L_0D_0L_0^{\mathrm{T}}$, i.e.,

$$\text{Is } \|\bigl(L_0D_0L_0^{\mathrm{T}} - \lambda_i I\bigr)z_i\| = \mathrm{O}(n\varepsilon \text{ spdiam } [T_0]) \quad \text{for all } i \in \Gamma_0?$$

Here spdiam$[T_0]$ is the spectral diameter of T_0.

We will need the following technical lemma that is of some interest in its own right. The proof uses two less than obvious results; one is the Demmel–Kahan result on tridiagonals with zero diagonal [4], the second concerns minimizing the spectral diameter.

Lemma 4. *Let* $(T = LDL^{\mathrm{t}}, \Gamma)$ *be an internal node, with* T *irreducible, in the representation tree satisfying Property* II *given in Section* 3.1 *namely*

$$\|D\boldsymbol{z}\| \leqslant C \text{ spdiam}[T_0], \quad \|\overset{o}{L}\, D\, \overset{o}{L}{}^{\mathrm{t}}\, \boldsymbol{z}\| \leqslant C \text{ spdiam}[T_0], \quad (T_0, \Gamma_0) \textit{ is the root}$$

for all computed normalized eigenvectors $\boldsymbol{z}$ *associated with* Γ (*recall that* $\overset{o}{L} = L - I$). *Consider small perturbations* $l_i \longrightarrow l_i(1 + \eta_i)$, $d_i \longrightarrow d_i(1 + \varepsilon_i)$, *where* $|\eta_i| \leqslant \bar{\eta}$, $|\varepsilon_i| \leqslant \bar{\varepsilon}$. *Write the perturbed matrix as* $T + \delta T$. *Then, for all the* $\boldsymbol{z}$,

$$\|\delta T\, \boldsymbol{z}\| \leqslant (2C + 1/2)(\bar{\varepsilon} + \bar{\eta}) \text{ spdiam}[T_0] + \mathrm{O}(n\varepsilon^2),$$

where $\max\{\bar{\varepsilon}, \bar{\eta}\} = \mathrm{O}(\varepsilon)$ *and* ε *is the roundoff unit.*

Proof. Let $D = \mathrm{diag}(d_1, \ldots, d_n)$ and $L_{i+1,i} = l_i$, so $\alpha_i := T_{ii} = d_i + d_{i-1}l_{i-1}^2$ and $\beta_i := T_{i,i+1} = d_il_i$. Observe that

$$\begin{aligned}
\delta T_{i,i+1} &= d_il_i(1 + \eta_i)(1 + \varepsilon_i) - \beta_i = \beta_i[\varepsilon_i + \eta_i + \mathrm{O}(\varepsilon^2)],\\
\delta T_{i+1,i+1} &= d_{i+1}(1 + \varepsilon_{i+1}) + d_il_i^2(1 + \varepsilon_i)(1 + \eta_i)^2 - \alpha_{i+1} \quad \text{for } i \geqslant 1,\\
&= d_{i+1}\varepsilon_{i+1} + d_il_i^2[\varepsilon_i + 2\eta_i + \mathrm{O}(\varepsilon^2)],
\end{aligned}$$

and

$$\delta T_{1,1} = (1 + \varepsilon_1)d_1 - \alpha_1 = \varepsilon_1 d_1.$$

For any unit vector $\boldsymbol{u}$,

$$\begin{aligned}
(\delta T\boldsymbol{u})_{i+1} &= \beta_iu_i(\varepsilon_i + \eta_i) + \beta_{i+1}u_{i+2}(\varepsilon_{i+1} + \eta_{i+1})\\
&\quad + d_{i+1}\varepsilon_{i+1}u_{i+1} + d_il_i^2(\varepsilon_i + 2\eta_i)u_{i+1} + \mathrm{O}(\varepsilon^2),\\
(\delta T\boldsymbol{u})_1 &= d_1\varepsilon_1u_1 + \beta_1u_2(\varepsilon_1 + \eta_1 + \varepsilon_1\eta_1).
\end{aligned}$$

Let $|M|$ denote the matrix of absolute values $(|m_{ij}|)$. Then

$$\|\delta T\boldsymbol{u}\| \leqslant (\bar{\varepsilon}+\bar{\eta})\||T-\mathrm{diag}(T)||\boldsymbol{u}|\| + \bar{\varepsilon}\|D\boldsymbol{u}\| + (\bar{\varepsilon}+2\bar{\eta})\|\overset{o}{L}\,D\,\overset{o}{L}{}^{\mathrm{t}}\,\boldsymbol{u}\| + \mathrm{O}(\varepsilon^2). \tag{23}$$

Since T is symmetric tridiagonal

$$\||T-\mathrm{diag}(T)|\| = \|T-\mathrm{diag}(T)\| = \tfrac{1}{2}\mathrm{spdiam}[T-\mathrm{diag}(T)].$$

The next task is to relate $T-\mathrm{diag}(T)$ to $T_0-\mathrm{diag}(T_0)$. In exact arithmetic they are identical because each representation LDL^{T} is a translate of T_0. Suppose LDL^{t} is computed from $L_1D_1L_1^{\mathrm{t}}$ (using the differential qd transform represented by Fig. 6). Then by analyzing the rounding errors (see Theorem 2 in [6]), $(LDL^{\mathrm{t}})_{i,i+1} = (L_1D_1L_1^{\mathrm{t}})_{i,i+1}(1+\xi_i)$, $|\xi_i| \leqslant 6\varepsilon$ where ε is the roundoff unit. Note that this relationship holds even if there is element growth in computing LDL^{t}. So, at a node with separation ν from the root we have, element by element,

$$|T-\mathrm{diag}(T)| \leqslant (1+\varepsilon)^{6\nu}|T_0-\mathrm{diag}(T_0)|.$$

The seminal paper [4] showed that a symmetric tridiagonal matrix with zero diagonal determines all of its eigenvalues to high relative accuracy; the relative change to any eigenvalue is bounded by the product of the relative changes to all the off-diagonal entries. In particular, the above inequality and Corollary 1 of Theorem 2 in [4] together imply that for each positive eigenvalue λ,

$$\lambda[T-\mathrm{diag}(T)] \leqslant ((1+\varepsilon)^{6\nu})^{n-1}\lambda[T_0-\mathrm{diag}(T_0)].$$

Since $\lambda_n[T-\mathrm{diag}(T)] = -\lambda_1[T-\mathrm{diag}(T)]$ where the ordering is $\lambda_1 \leqslant \cdots \leqslant \lambda_n$,

$$\mathrm{spdiam}[T-\mathrm{diag}(T)] \leqslant (1+\varepsilon)^{(n-1)6\nu}\mathrm{spdiam}[T_0-\mathrm{diag}(T_0)]. \tag{24}$$

Now we can bound the first term in (6). Since $\boldsymbol{u}$ is a unit vector,

$$\begin{aligned}\||T-\mathrm{diag}(T)||\boldsymbol{u}|\| &\leqslant \|T-\mathrm{diag}(T)\| = \tfrac{1}{2}\mathrm{spdiam}[T-\mathrm{diag}(T)]\\ &\leqslant \tfrac{1}{2}(1+\varepsilon)^{(n-1)6\nu}\mathrm{spdiam}[T_0-\mathrm{diag}(T_0)], \quad \text{by (24)}.\end{aligned}$$

Replace $\boldsymbol{u}$ by the computed vector $\boldsymbol{z}$, invoke the hypothesis (Property II) for the second and third terms in (6) to find

$$\begin{aligned}\|\delta T\boldsymbol{z}\| \leqslant{}& \tfrac{1}{2}(\bar{\varepsilon}+\bar{\eta})(1+\varepsilon)^{(n-1)6\nu}\mathrm{spdiam}[T_0-\mathrm{diag}(T_0)] + \bar{\varepsilon}C\,\mathrm{spdiam}[T_0]\\ &+ (\bar{\varepsilon}+2\bar{\eta})C\,\mathrm{spdiam}[T_0] + \mathrm{O}(\varepsilon^2).\end{aligned}$$

Finally Theorem 2 in [11], applied to the irreducible symmetric tridiagonal T_0, gives

$$\mathrm{spdiam}[T_0-\mathrm{diag}(T_0)] \leqslant \mathrm{spdiam}[T_0],$$

so

$$\|\delta T z\| \leqslant (\bar{\varepsilon} + \bar{\eta})\,\mathrm{spdiam}[T_0](2C + (1/2)(1+\varepsilon)^{6\nu n}).$$

Specific values will be given to $\bar{\eta}$ and $\bar{\varepsilon}$ in Theorem 4. □

6.1. Propagation of residual norms

Recall that we designate eigenvalues by indices because the actual values change with the shift. Consider a typical unit vector z_k computed at some leaf node by Algorithm Getvec. By (19), z_k is extremely close to an eigenvector $\boldsymbol{v}_k$ of the parent LDL^{t} of that leaf node. Thus the residual norm $\|(LDL^{\mathrm{t}} - \{k\}I)z_k\|$ is small for the parent of a leaf node. However our goal is to bound $\|(L_0D_0L_0^{\mathrm{t}} - \{k\}I)z_k\|$ where $L_0D_0L_0^{\mathrm{t}}$ is the root representation.

Theorem 4. *Let $T_0 = L_0D_0L_0^{\mathrm{T}}$ be the initial representation for which Algorithm* MR^3 *computes eigenvectors z_k, $k \in \Gamma_0$, the initial index set. Assume that all internal nodes* $(LDL^{\mathrm{T}}, \Gamma)$ *satisfy Property* II *in Section* 3.1 *and that Algorithm* Getvec *computes z_k such that*

$$\sin \angle(z_k, \boldsymbol{v}_k) \leqslant \mathscr{G} n\varepsilon, \tag{25}$$

where $\boldsymbol{v}_k$ is the corresponding eigenvector of $\{k\}$*'s parent node* (*see* (19)). *Then the residual norm satisfies*

$$\|(T_0 - \hat{\lambda}_k I)z_k\| \leqslant \mathscr{G} n\varepsilon + 9(\mathrm{ndepth} - 1)(2C + 1/2)\varepsilon\,\mathrm{spdiam}[T_0] + \mathrm{O}(n\varepsilon^2), \tag{26}$$

where ndepth *is the depth of the representation tree and $\hat{\lambda}_k$ is the value* (*with respect to the root T_0*) *of the kth computed eigenvalue.*

Proof. We will obtain this bound by following the residual norm up the tree. Our tool on ascending an edge will be the commutative diagram in Fig. 6 for the parent to child transformation

$$LDL^{\mathrm{t}} - \tau I \longrightarrow L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}},$$

where LDL^{t} and $L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}$ are intermediate representations formed in computing z_k.

Let $\hat{\lambda}$ be the value of the computed eigenvalue $\{k\}$ for the representation $L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}$, $k \in \Gamma_{\mathrm{c}}$. Corresponding to the four matrices in Fig. 6 we have four residual norms,

$$\|\boldsymbol{r}_{\mathrm{c}}\| := \|(L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}} - \hat{\lambda}I)z_k\|,$$
$$\|\bar{\boldsymbol{r}}_{\mathrm{c}}\| := \|(\bar{L}_{\mathrm{c}}\bar{D}_{\mathrm{c}}\bar{L}_{\mathrm{c}}^{\mathrm{t}} - \hat{\lambda}I)z_k\|,$$

$$\|\bar{\boldsymbol{r}}\| := \|(\bar{L}\bar{D}\bar{L}^{\mathrm{t}} - (\hat{\lambda} + \tau)I)z_k\|,$$

and

$$\|\boldsymbol{r}\| := \|(LDL^{\mathrm{t}} - (\hat{\lambda} + \tau)I)z_k\|.$$

By the exact relation $\bar{L}\bar{D}\bar{L}^{\mathrm{t}} - \tau = \bar{L}_{\mathrm{c}}\bar{D}_{\mathrm{c}}\bar{L}_{\mathrm{c}}^{\mathrm{t}}$ we have

$$\|\bar{\boldsymbol{r}}_{\mathrm{c}}\| = \|\bar{\boldsymbol{r}}\|. \tag{27}$$

With $\delta T_{\mathrm{c}} := \bar{L}_{\mathrm{c}}\bar{D}_{\mathrm{c}}\bar{L}_{\mathrm{c}}^{\mathrm{t}} - L_{\mathrm{c}}D_{\mathrm{c}}L_{\mathrm{c}}^{\mathrm{t}}$, and similarly for $\delta T := \bar{L}\bar{D}\bar{L}^{\mathrm{t}} - LDL^{\mathrm{t}}$,

$$\begin{aligned}
\|\bar{\boldsymbol{r}}_{\mathrm{c}}\| &= \|\boldsymbol{r}_{\mathrm{c}} + \delta T_{\mathrm{c}} z_k\| \leqslant \|\boldsymbol{r}_{\mathrm{c}}\| + \|\delta T_{\mathrm{c}} z_k\|,\\
\|\boldsymbol{r}\| &= \|\bar{\boldsymbol{r}} + \delta T z_k\| \leqslant \|\bar{\boldsymbol{r}}\| + \|\delta T z_k\|.
\end{aligned}$$

So, using (27),

$$\|\boldsymbol{r}\| \leqslant \|\boldsymbol{r}_{\mathrm{c}}\| + \|\delta T_{\mathrm{c}} z_k\| + \|\delta T z_k\|. \tag{28}$$

Both δT_{c} and δT have the structure determined by the error analysis of the dstqds transform. On the path from leaf to root each internal node contributes twice to the augmentation of the residual norm, once as a parent and once as a child; the two perturbations differ but the bound on them is very close. The ulp changes in Fig. 6 show that

As a parent: $\bar{\varepsilon} = 1 \cdot \varepsilon, \quad \bar{\eta} = 3 \cdot \varepsilon,$
As a child: $\bar{\varepsilon} = 2 \cdot \varepsilon, \quad \bar{\eta} = 3 \cdot \varepsilon,$

where $\bar{\varepsilon}$ and $\bar{\eta}$ are as in Lemma 4. See [6, Section 5] for the error analysis. Substitute these values in Lemma 4 to find,

$$\begin{aligned}
\|\delta T z_k\| &\leqslant 4(2C + 1/2)\varepsilon \text{ spdiam } [T_0] + \mathrm{O}(n\varepsilon^2),\\
\|\delta T_{\mathrm{c}} z_k\| &\leqslant 5(2C + 1/2)\varepsilon \text{ spdiam}[T_0] + \mathrm{O}(n\varepsilon^2).
\end{aligned} \tag{29}$$

Now apply (28) at each internal node on the path from the leaf for eigenvalue $\{k\}$ to the root and invoke (29) to get

$$\begin{aligned}
\|(T_0 - \lambda_k I)z_k\| &\leqslant \|\boldsymbol{r}_{\text{initial}}\| + \sum_{\text{node}\in\text{path}} (\|\delta T_{\mathrm{c}} z_k\| + \|\delta T z_k\|),\\
&\leqslant \|\boldsymbol{r}_{\text{initial}}\| + (2C + 1/2)\varepsilon \sum_{\text{path}} 9 \text{ spdiam}[T_0] + \mathrm{O}(n\varepsilon^2),
\end{aligned} \tag{30}$$

the sum being over internal nodes on the path. Here $\boldsymbol{r}_{\text{initial}}$ is the residual at the leaf node with respect to the corresponding twisted factorization. By the proof of Theorem 15 in [6],

$$\|\boldsymbol{r}_{\text{initial}}\| \leqslant \mathscr{G} n\varepsilon$$

and thus by (30)

$$\|(T_0 - \lambda_k I)z_k\| \leqslant \mathscr{G} n\varepsilon + 9(\text{ndepth} - 1)(2C + 1/2)\varepsilon\ \text{spdiam}[T_0] + \text{O}(n\varepsilon^2),$$

where ndepth is the depth of the representation tree. □

7. Conclusions

In this paper, we have presented Algorithm MR^3 that computes k eigenvectors of a symmetric tridiagonal in $\text{O}(kn)$ time. The salient feature of the proposed algorithm is that multiple representations LDL^{t} are used, and each eigenvector is computed to high accuracy with respect to the appropriate representation. No Gram–Schmidt orthogonalization is needed.

Proving that the computed eigenvectors are numerically orthogonal and have small residual norms has been a major concern of this paper. Due to the multiple representations involved the proof is somewhat complicated. The proofs require that each representation be a relatively robust representation (RRR) for the eigenpairs that are to be computed using that representation. For the purpose of this paper, we have assumed that each representation is an RRR. There has been considerable work in showing the conditions under which RRRs exist, such as in [6,13], however this is beyond the scope of this paper. In practice, finding appropriate RRRs is easy and checkable; indeed Algorithm MR^3 has been realized as the software routine xSTEGR that is included in LAPACK [1].

References

[1] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, D. Sorensen, LAPACK Users' Guide, second ed., SIAM, Philadelphia, 1995, 324 p.

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Introduction to Algorithms, MIT Electrical and Computer Science Series, MIT Press, 1992, ISBN 0-262-03141-8.

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

[4] J. Demmel, W. Kahan, Accurate singular values of bidiagonal matrices, SIAM J. Sci. Stat. Comput. 11 (5) (1990) 873–912.

[5] I.S. Dhillon, A new $\text{O}(n^2)$ algorithm for the symmetric tridiagonal eigenvalue/eigenvector problem, Ph.D. Thesis, Computer Science Division, University of California, Berkeley, California, May 1997. Available as UC Berkeley Technical Report No. UCB//CSD-97-971.

[6] I.S. Dhillon, B.N. Parlett, Orthogonal eigenvectors and relative gaps, SIAM J. Matrix Anal. Appl. 25 (4) (2004), in press. Also LAPACK Working Note 154 (ut-cs-02-474).

[7] S. Eisenstat, I. Ipsen, Relative perturbation techniques for singular value problems, SIAM J. Numer. Anal. 32 (6) (1995).

[8] K. Fernando, B. Parlett, Accurate singular values and differential qd algorithms, Numer. Math. 67 (2) (1994) 191–229.

[9] Gene H. Golub, Charles F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, 1996.

[10] M. Gu, S.C. Eisenstat, A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem, SIAM J. Matrix Anal. Appl. 16 (1) (1995) 172–191.

[11] B.N. Parlett, The spectral diameter as a function of the diagonal entries, Numer. Linear Algebra Appl. 1 (2003) 1–7.

[12] B.N. Parlett, I.S. Dhillon, Fernando's solution to Wilkinson's problem: an application of double factorization, Linear Algebra Appl. 267 (1997) 247–279.

[13] B.N. Parlett, I.S. Dhillon, Relatively robust representations of symmetric tridiagonals, Linear Algebra Appl. 309 (2000) 121–151.

[14] H. Rutishauser, Der quotienten-differenzen-algorithmus, Z. Angew. Math. Phys. 5 (1954) 223–251.

[15] J. Rutter, A serial implementation of Cuppen's divide and conquer algorithm for the symmetric eigenvalue problem, Mathematics Dept. Master's Thesis, University of California, 1994.

[16] J.H. Wilkinson, The Algebraic Eigenvalue Problem, Oxford University Press, Oxford, 1965.