Available online at www.sciencedirect.com
ScienceDirect
Computers and Mathematics with Applications 56 (2008) 166–171
An International Journal
computers & mathematics with applications
www.elsevier.com/locate/camwa

Eigen-solving via reduction to DPR1 matrices[☆]

V.Y. Pan[a,*], B. Murphy[a], R.E. Rosholt[a], Y. Tang[b], X. Wang[c], A. Zheng[d]

[a] *Department of Mathematics and Computer Science, Lehman College, City University of New York, Bronx, NY 10468, USA*
[b] *Ph.D. Program in Computer Science, The City University of New York, New York, NY 10036, USA*
[c] *Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, China*
[d] *Ph.D. Program in Mathematics, The City University of New York, New York, NY 10036, USA*

Received 7 August 2007; accepted 18 September 2007

Abstract

Highly effective polynomial root-finders have been recently designed based on eigen-solving for DPR1 (that is diagonal + rank-one) matrices. We extend these algorithms to eigen-solving for the general matrix by reducing the problem to the case of the DPR1 input via intermediate transition to a TPR1 (that is triangular + rank-one) matrix. Our transforms use substantially fewer arithmetic operations than the QR classical algorithms but employ non-unitary similarity transforms of a TPR1 matrix, whose representation tends to be numerically unstable. We, however, operate with TPR1 matrices implicitly, as with the inverses of Hessenberg matrices. In this way our transform of an input matrix into a similar DPR1 matrix partly avoids numerical stability problems and still substantially decreases arithmetic cost versus the QR algorithm.

Keywords: The algebraic eigenproblem; DPR1 matrices; TPR1 matrices; Hessenberg matrices

1. Introduction

Hereafter we use the abbreviations “DPR1” for “diagonal + rank-one”, “TPR1” for “triangular + rank-one”, and “ops” for “real arithmetic operations”. We use letters in boldface to denote vectors and use capital letters to denote matrices. M^{T} and $\mathbf{v}^{\mathrm{T}}$ denote the transposes of a matrix M and a vector $\mathbf{v}$, respectively. I is the $n \times n$ identity matrix, and $\mathbf{e}_i$ is its ith column for $i = 1, \ldots, n$. For simplicity we restrict our study to real matrices, but the extension to the complex case is routine.

Highly effective eigen-solvers (using $O(n^2)$ ops overall) have been recently devised for DPR1 matrices, motivated by the equivalence of this task to polynomial root-finding [1–5]. At first the paper [1] proposed and analyzed an effective eigen-solver for a DPR1 and Frobenius matrices based on the inverse power iteration. The other cited papers have followed this pattern of exploiting matrix structure for polynomial root-finding but replaced the structured inverse iteration with structured versions of the QR algorithm. It is still unclear which variant of the approach is most powerful, possibly the third one based on incorporating unsymmetric Lanczos algorithm.

[☆] Supported by PSC CUNY Awards 67297–0036 and 68291–0037.
[*] Corresponding author.
E-mail addresses: victor.pan@lehman.cuny.edu (V.Y. Pan), brian.murphy@lehman.cuny.edu (B. Murphy), rhys.rosholt@lehman.cuny.edu (R.E. Rosholt), ytang@gc.cuny.edu (Y. Tang), xinmao@ustc.edu.cn (X. Wang), azheng1999@yahoo.com (A. Zheng).
URL: http://comet.lehman.cuny.edu/vpan/ (V.Y. Pan).

0898-1221/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2007.09.012

In this paper we examine the natural direction of extending the cited advance in eigen-solving to a more general class of matrices by obtaining their similarity transforms into DPR1 matrices. In [6] non-unitary similarity transforms TPR1 $\rightarrow$ DPR1 have been proposed, and it was shown that there exist no real unitary similarity transforms of this kind.

Theoretically the paper [6] has extended effective known eigen-solvers for DPR1 matrices to the case of any input matrix because we have unitary similarity transforms of any matrix into either a block triangular or a TPR1 matrix [7]. The transforms in [7] use about as many ops as the classical similarity transform of a matrix into the (upper) Hessenberg form [8].

Practically, however, one can expect to have numerical stability problems due to using non-unitary transforms of TPR1 matrices in [6]. In the present paper we extend the unitary stage by obtaining a TPR1 matrix with a nondefective triangular term. The remaining non-unitary stage is limited to computing the eigenvectors of this term whose eigenvalues (equal to the diagonal entries) we can choose at will.

Additional care must be taken about TPR1 representation of the auxiliary matrices, which can also cause numerical problems. The remedies in [2,9,10] are not much costly for DPR1 matrices, defined by $3n-1$ parameters, but become a more serious burden for a TPR1 matrix, which depends on $(n+1)n/2+2n-1$ parameters. To alleviate these difficulties we modify the TPR1 $\rightarrow$ DPR1 transforms by representing the TPR1 matrices implicitly, as the inverses of Hessenberg matrices.

As a whole our work is preliminary, with the main goal of giving some new insights, showing new techniques, and motivating further effort.

We organize our presentation as follows. In the next section we describe our transform from a TPR1 matrix into a nondefective TPR1 matrix. In Section 3 we revisit eigen-solving for a nondefective triangular input and extend it to a transform TPR1 $\rightarrow$ DPR1. In short Section 4 we compare the arithmetic cost of this transform and the QR algorithm. In Section 5 we extend the classical transform of a matrix into a Hessenberg matrix further, to a DPR1 matrix. We leave Section 6 for a discussion.

2. How to ensure nondefectiveness

In this section we devise a unitary similarity transform of a TPR1 matrix $R+\mathbf{u}\mathbf{v}^{\mathrm{T}}$ into a TPR1 matrix $\tilde{R}+\tilde{\mathbf{u}}\tilde{\mathbf{v}}^{\mathrm{T}}$ with a nondefective upper triangular term $\tilde{R}$. We recursively alternate the stages of *TPR1 rearrangement* and *reordering the eigenvalues of a triangular matrix*.

1. *TPR1 rearrangement*

For our input TPR1 matrix $T=R+\mathbf{u}\mathbf{v}^{\mathbf{T}}$, $R=(r_{ij})_{i,j=1}^{n}$, $\mathbf{u}=(u_i)_{i=1}^{n}$, $\mathbf{v}=(v_j)_{j=1}^{n}$, we assume that $u_n v_1 \neq 0$. Otherwise the matrix T would have had all its subdiagonal entries in the first column and/or the last row equal to zero and thus would have had the eigenvalues r_{11} and/or r_{nn}. We can readily deflate such a matrix.

Now, for every pair of scalars a and b we have $R+\mathbf{u}\mathbf{v}^{\mathrm{T}}=R^{(1)}+\mathbf{u}^{(1)}\mathbf{v}^{(1)\mathrm{T}}$ where $\mathbf{u}^{(1)}=\mathbf{u}-a\mathbf{e}_1$, $\mathbf{v}^{(1)}=\mathbf{v}-b\mathbf{e}_n$, and $R^{(1)}=R+a\mathbf{e}_1\mathbf{v}^{\mathrm{T}}+b\mathbf{u}\mathbf{e}_n^{\mathrm{T}}-ab\mathbf{e}_1\mathbf{e}_n^{\mathrm{T}}=R+a\mathbf{e}_1\mathbf{v}^{(1)\mathrm{T}}+b\mathbf{u}\mathbf{e}_n$. We observe that $R^{(1)}$ is an upper triangular matrix with the eigenvalues $r_{ii}^{(1)}=r_{ii}$ for $i=2,3,\ldots,n-1$, $r_{11}^{(1)}=r_{11}+av_1$, $r_{nn}^{(1)}=r_{nn}+bu_n$.

By choosing appropriate scalars a and b we can obtain any pair of entries $r_{11}^{(1)}$ and $r_{nn}^{(1)}$. We let these entries be distinct from each other and all other diagonal entries $r_{ii}^{(1)}$. The computation of the matrix $R^{(1)}$ and the vectors $\mathbf{u}^{(1)}$ and $\mathbf{v}^{(1)}$ takes $4n+4$ ops. The bound decreases by twice if $ab=0$. In both cases we need $2n+2$ ops per a diagonal entry.

2. *Reordering the eigenvalues of a triangular matrix*

We interchange the order of two distinct adjacent diagonal entries of a triangular matrix by applying Givens rotations [8, Section 7.6], which define the following transforms for 2×2 diagonal blocks, $\begin{pmatrix} r_1 & t \\ 0 & r_2 \end{pmatrix} \rightarrow \begin{pmatrix} r_2 & \pm t \\ 0 & r_1 \end{pmatrix}$.

Such a reordering stage takes $12n$ ops per block.

By recursively applying Stages 1 and 2 we obtain a robust unitary similarity transform $R+\mathbf{u}\mathbf{v}^{\mathrm{T}} \rightarrow \tilde{R}+\tilde{\mathbf{u}}\tilde{\mathbf{v}}^{\mathrm{T}}=Q(R+\mathbf{u}\mathbf{v}^{\mathrm{T}})Q^{\mathrm{T}}$ where Q denotes a unitary matrix and all eigenvalues $r_{ii}^{(1)}=\tilde{r}_{ii}$ of the triangular matrix $\tilde{R}$ are distinct, so that the matrix is nondefective.

In the worst case, where the matrix R has single eigenvalue of multiplicity n, we apply the above transform to modify $n-1$ diagonal entries. Each modification takes $2n+2$ ops, which means $2n^2-2$ ops in all TPR1 rearrangements. In this case we also need either k^2 reorderings and therefore $12k^2n = 3(n-1)^2n$ ops for odd $n = 2k+1$ or $(k-1)k$ reorderings and therefore $12(k-1)kn = 3(n-2)n^2$ ops for even $n = 2k$. In both cases we use $3n^3 + O(n^2)$ ops overall.

3. TPR1 – DPR1 transform

Given a TPR1 $\to$ DPR1 for a TPR1 input matrix $T = R + \mathbf{u}\mathbf{v}^T$ with a nondefective triangular term R, we first compute all its eigenvectors and then readily transform it into a similar DPR1 matrix.

Algorithm 3.1. TPR1 $\to$ DPR1 transform.

1. Compute and output a nonsingular matrix Y of the right eigenvectors of the matrix R such that $D = XRY = \mathrm{diag}(r_{ii})_{i=1}^n$ for $X = Y^{-1}$ is the diagonal matrix that shares its diagonal entries with the matrix R.
2. Compute and output the vectors $\mathbf{s} = X\mathbf{u}$ and $\mathbf{t}^T = \mathbf{v}^T Y$.

The algorithm defines the desired rational similarity transform $X(R + \mathbf{u}\mathbf{v}^T)Y = D + \mathbf{s}\mathbf{t}^T$.

Theorem 3.2. *Algorithm 3.1 can be performed by using $(2n^2 + 15n - 5)n/3$ ops.*

Proof. It is sufficient to compute a matrix Y of the right eigenvectors of the triangular term R and the vectors $\mathbf{s} = X\mathbf{u}$ and $\mathbf{t}^T = Y\mathbf{v}^T$. Compute the matrix Y as a lower triangular matrix whose ith column vector $\mathbf{y}_i$ has the form $(\mathbf{0}_{n-i}^T, \mathbf{z}_i^T)^T$ where the subvector $\mathbf{z}_i$ is in the null space of the $i \times i$ trailing principal submatrix R_i of the matrix $R - r_{ii}I$, that is, $R_i\mathbf{z}_i = \mathbf{0}_i$, $i = 1, \dots, n$. Set the last coordinate of this vector equal to one and compute the other nonzero coordinates in $i^2 - i$ ops with the back substitution. For $i = 1, \dots, n$ this computation produces a desired matrix Y in $(2n+1)(n+1)n/6 - (n+1)n/2 = (n^3 - n)/3$ ops. We also use $(n-1)n/2$ subtractions to compute the scalars $r_{ii} - r_{jj}$ for $j = 1, \dots, i-1$; $i = 1, \dots, n$. It remains to compute the vectors $\mathbf{s}$ and $\mathbf{t}$. At this stage $2n^2$ ops are sufficient. Similarly we can compute the matrix X whose ith row vector has the form $(\mathbf{w}^{(i)T}, \mathbf{0}_{n-i}^T)^T$ where the subvector $\mathbf{w}^{(i)T}$ lies in the null space of the $i \times i$ leading principal submatrix $R^{(i)}$ of the matrix $R - r_{ii}I$, that is, $\mathbf{w}^{(i)T}R^{(i)} = \mathbf{0}_i^T$, $i = 1, \dots, n$. Summarizing, we obtain the theorem. □

Remark 3.3. In fact we can save at least the order of $n^3/12$ ops if we compute the $n/2 \times n/2$ leading and trailing blocks of the matrix R^{-1} via inverting the respective blocks of the matrix R (cf. Block diagonalization redux in [11, Section 1.1.5]).

Algorithm 3.1 can be applied whenever the matrix R is nondefective. We observe some freedom in devising unitary similarity transforms into TPR1 matrices in the nonsymmetric version of the algorithm in [12].

4. Comparison with the arithmetic cost of the QR algorithm

Let us compare the estimated arithmetic cost of our real unitary similarity transform $R + \mathbf{u}\mathbf{v}^T \to \tilde{R} + \tilde{\mathbf{u}}\tilde{\mathbf{v}}^T$ with the respective estimates for the QR algorithm. The other stages of our algorithm use about as many ops as the respective stages of the QR algorithms. Namely, the reduction to the TPR1 form in [7] is roughly as costly as the classical reduction to the Hessenberg form, and one can diagonalize the triangular matrix R in $(2n^2 + 15n - 5)n/3$ ops in both eigen-solvers (that is the QR iteration and our transform). Finally eigen-solving for a DPR1 matrix requires $O(n^2)$ ops.

Now, we recall that our transform $R + \mathbf{u}\mathbf{v}^T \to \tilde{R} + \tilde{\mathbf{u}}\tilde{\mathbf{v}}^T$ requires $3n^3 + O(n^2)$ ops (see Section 2), whereas the QR algorithm uses $20kn^3 + O(kn^2)$ ops where $k \geq 2$ is the observed number of QR iterations required per an eigenvalue [11, Section 2.3.3].

5. Inverse Hessenberg → DPR1 transform

5.1. Flowchart

Next recall that the inverse of a nonsingular Hessenberg matrix H having no zeros on the first subdiagonal is equal to a TPR1 matrix whose diagonal is filled with zeros [13]. This enables an alternative to the transform in [7], that is we can apply the classical unitary transform of an input matrix into a similar Hessenberg matrix H and, if needed, apply shifts to ensure its nonsingularity or even diagonal dominance. It remains to transform the inverses of the resulting Hessenberg matrix or its Hessenberg diagonal blocks into DPR1 matrices. The original eigenproblem has been reduced to the case of the TPR1 matrix $H^{-1} = T = R + \mathbf{u}\mathbf{v}^{\mathrm{T}}$, implicitly defined by the Hessenberg matrix H. Now we propose the following flowchart.

Flowchart 5.1. Inverse Hessenberg → DPR1 transform.

1. Compute the vectors $\mathbf{u}$ and $\mathbf{v}$ from the linear systems $aH\mathbf{u} = \mathbf{e}_1$ and $H^{\mathrm{T}}\mathbf{v} = \mathbf{e}_n$ choosing the scalar a so that $u_n = 1$ for $\mathbf{u} = (u_i)_{i=1}^n$.
2. Apply the unitary similarity transform in Section 2 to transform the TPR1 input matrix H^{-1} into a TPR1 matrix $QH^{-1}Q^{\mathrm{T}} = R + \mathbf{u}\mathbf{v}^{\mathrm{T}}$ that has a nondefective triangular term R. (To simplify the notation we write R, $\mathbf{u}$, and $\mathbf{v}$ rather than $\tilde{R}$, $\tilde{\mathbf{u}}$, and $\tilde{\mathbf{v}}$.)
3. Compute two nonsingular upper triangular matrices $Y = (\mathbf{y}_i)_{i=1}^n$ and $X = Y^{-1}$ such that $XRY = D$ is a diagonal matrix, and so $X(R + \mathbf{u}\mathbf{v}^{\mathrm{T}})Y = D + \mathbf{s}\mathbf{t}^{\mathrm{T}}$ is a DPR1 matrix where $\mathbf{s} = X\mathbf{u}$ and $\mathbf{t} = Y^{\mathrm{T}}\mathbf{v}$.
4. Use the latter expressions to compute the vectors $\mathbf{s}$ and $\mathbf{t}$.
5. Apply the algorithms in [1–5] to solve the eigenproblem for the DPR1 matrix $D + \mathbf{s}\mathbf{t}^{\mathrm{T}}$.
6. Recover the solution of the eigenproblem for the original matrix H^{-1}.

5.2. *Elaboration and modifications*

At Stage 3 of the flowchart we can apply the algorithm of Section 3, but next we show two alternative approaches where we operate with the matrix $QH^{-1}Q^{\mathrm{T}} - \mathbf{u}\mathbf{v}^{\mathrm{T}}$.

1. The inverse power iteration applied to the matrix R with the shifts $r_i \approx r_{ii}$ for all i recursively updates the pair of initial approximate eigenvalue $r_i \approx r_{ii}$ and eigenvector $\mathbf{y}_i$ according to the relations

$$\mathbf{y}_i \leftarrow (R - r_i I)^{-1}\mathbf{y}_i / \|(R - r_i I)^{-1}\mathbf{y}_i\|,$$
$$r_i \leftarrow \mathbf{y}_i^{\mathrm{T}} R \mathbf{y}_i.$$

Initially we can choose, any $r_i \approx r_{ii}$ such that $r_i \neq r_{ii}$ and, e.g., $\mathbf{y}_i = \mathbf{e}_1$.

By replacing the input matrix $R \leftarrow QH^{-1}Q^{\mathrm{T}} - \mathbf{u}\mathbf{v}^{\mathrm{T}}$, we rewrite the iteration as follows,

$$\mathbf{y}_i \leftarrow (QH^{-1}Q^{\mathrm{T}} - \mathbf{u}\mathbf{v}^{\mathrm{T}} - r_i I)^{-1}\mathbf{y}_i / \|(QH^{-1}Q^{\mathrm{T}} - \mathbf{u}\mathbf{v}^{\mathrm{T}} - r_i I)^{-1}\mathbf{y}_i\|, \tag{5.1}$$
$$r_i \leftarrow \mathbf{y}_i^{\mathrm{T}}(QH^{-1}Q^{\mathrm{T}} - \mathbf{u}\mathbf{v}^{\mathrm{T}})\mathbf{y}_i. \tag{5.2}$$

2. By applying the recent modification of the inverse iteration based on additive preconditioning in [14–16], we simplify updating in (5.1) as follows,

$$\mathbf{y}_i \leftarrow (QH^{-1}Q^{\mathrm{T}} - r_i I)^{-1}\mathbf{u} / \|(QH^{-1}Q^{\mathrm{T}} - r_i I)^{-1}\mathbf{u}\|. \tag{5.3}$$

Since $R - r_i I = QH^{-1}Q^{\mathrm{T}} - \mathbf{u}\mathbf{v}^t - r_i I$, it can be proved (cf. [14]) that the transition to the recursive process (5.2) and (5.3) preserves quadratic convergence of iteration (5.1) and (5.2). Furthermore in this modification the choice $r_i = r_{ii}$ (prohibitive in the inverse iteration) is allowed and moreover enables instant convergence in a single step. (Numerically we also have such an instant convergence if r_i sufficiently closely approximates the eigenvalue r_{ii} of the matrix R.)

Suppose iterations (5.1) and (5.2) or (5.1) and (5.3) have output the simple eigenvalue r_{ii} and the associated right eigenvector $\mathbf{y}_i$. Then we can immediately compute the associated left eigenvector $\mathbf{x}_i$, for which we have the following expression [14–16],

$$\mathbf{x}_i = \mathbf{v}^{\mathrm{T}}(QH^{-1}Q^{\mathrm{T}} - r_i I)^{-1} / \|\mathbf{v}^{\mathrm{T}}(QH^{-1}Q^{\mathrm{T}} - r_i I)^{-1}\|. \tag{5.4}$$

One can avoid involving the inverse H^{-1} in expressions (5.1)–(5.4) by substituting the equations

$$(QH^{-1}Q^{\mathrm{T}} - r_i I)^{-1} = QH(I/r_i - H)^{-1}Q^{\mathrm{T}}/r_i \tag{5.5}$$

and

$$(QH^{-1}Q^{\mathrm{T}} - \mathbf{u}\mathbf{v}^{\mathrm{T}} - r_i I)^{-1} = QH(I - r_i H - Q^{\mathrm{T}}\mathbf{u}\mathbf{v}^{\mathrm{T}}QH)^{-1}Q^{\mathrm{T}}. \tag{5.6}$$

5.3. Computational cost

Let us estimate the arithmetic cost of computations according to expressions (5.2)–(5.6) for all i. We need $\sum_{j=1}^{n}(2n-j) = (3n-1)n/2$ ops to factorize an $n \times n$ Hessenberg matrix into the product of a bidiagonal and triangular matrix, that is, $1.5\,n^3 + O(n^2)$ ops for all i. The subsequent solution of linear systems (5.3) and (5.4) takes $2n^2$ ops for a fixed i, which means $2n^3$ ops for all i. We should also include the $10n^3 + O(n^2)$ ops for multiplication of the matrices Q^{T}, H, and Q by vectors and $3n^3 + O(n^2)$ ops for ensuring nondefectiveness of the triangular term R (see Section 2). Summarizing we need $16.5n^3 + O(n^2)$ ops overall. This is still less than $20kn^3 + O(kn^2)$ ops expected in the QR algorithm.

6. Discussion

As we commented in the introduction, our work is preliminary, with the main goal of giving some new insights, showing new techniques, and motivating further theoretical and experimental study. In particular one should assess numerical problems that can occur in our non-unitary transforms and in using TPR1 representations.

Recall that we have avoided explicit TPR1 representation in the inverse Hessenberg → DPR1 transform (see Section 5.2), but we still use it in Section 2. Is it possible to avoid it also there? One can apply, e.g., the algorithm of [12] to a Hessenberg matrix H to obtain its unitary similarity transform into a TPR1 matrix $R + \mathbf{u}\mathbf{v}^{\mathrm{T}} = QHQ^{\mathrm{T}}$ whose triangular term R has n distinct diagonal entries. The subsequent transform into a DPR1 matrix can rely on Eqs. (5.2)–(5.4) simplified by replacing H^{-1} with H.

Here are some other open issues.

1. One should re-examine and adjust the known recipes for treating the harder case where the Hessenberg matrix H has small but nonzero subdiagonal entries (cf. [8,11]).

2. It is attractive to choose absolutely large shift values s to ensure a strong diagonal dominance of the shifted Hessenberg matrix $H = \tilde{H} - sI$. It is not yet clear, however, which shift would diminish numerical problems at the stage of computing the matrices $X = Y^{-1}$ and Y of the left and right eigenvectors of the matrix $R = H^{-1} - \mathbf{u}\mathbf{v}^{\mathrm{T}}$.

3. The TPR1 representation $R + \mathbf{u}\mathbf{v}^{\mathrm{T}}$ of the matrix $T = H^{-1}$ is not unique (cf. Section 2), and we may try to exploit such nonuniqueness to facilitate our computations.

References

[1] D.A. Bini, L. Gemignani, V.Y. Pan, Inverse power and Durand/Kerner iteration for univariate polynomial root-finding, Computers and Mathematics (with Applications) 47 (2–3) (2004) 447–459. Also Technical Reports 2002003 and 2002020, CUNY Ph.D. Program in Computer Science, Graduate Center, City University of New York, 2002.

[2] D.A. Bini, L. Gemignani, V.Y. Pan, Fast and stable QR eigenvalue algorithms for generalized companion matrices and secular equation, Numerische Mathematik 3 (2005) 373–408. Also Technical Report 1470, Department of Math., University of Pisa, Pisa, Italy, July 2003.

[3] D.A. Bini, L. Gemignani, V.Y. Pan, Improved initialization of the accelerated and robust QR-like polynomial root-finding, Electronic Transactions on Numerical Analysis 17 (2004) 195–205.

[4] V.Y. Pan, D. Ivolgin, B. Murphy, R.E. Rosholt, Y. Tang, X. Wang, X. Yan, Root-finding with eigen-solving, in: Dongming Wang, Lihong Zhi (Eds.), Symbolic–Numerical Computation, Birkhäuser, Basel, Boston, 2007, pp. 219–245.

[5] L. Gemignani, Structured matrix method for polynomial root-finding, in: Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC'07, ACM Press, New York, 2007, pp. 175–180.

[6] V.Y. Pan, M. Kunin, B. Murphy, R.E. Rosholt, Y. Tang, X. Yan, W. Cao, Linking the TPR1, DPR1, and arrow-head matrix structure, Computers and Mathematics (with Applications) 52 (2006) 1603–1608.

[7] R. Vandebril, M. Van Barel, N. Mastronardi, An orthogonal similarity reduction of a matrix to a semiseparable form, Technical Report TW353, Katholieke Universiteit Leuven, Department Computerwetenschappen, Leuven, Belgium, 2003.

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

[9] Y. Eidelman, I. Gohberg, Fast inversion algorithms for a class of block structured matrices, Contemporary Mathematics 281 (2001) 17–38.

[10] R. Vandebril, M. Van Barel, N. Mastronardi, A note on the representation and definition of semiseparable matrices, Numerical Linear Algebra with Applications 12 (2005) 839–858. Also Technical Report TW393, Katholieke Universiteit Leuven, Department Computerwetenschappen, Leuven, Belgium, 2004.

[11] G.W. Stewart, Matrix Algorithms, Vol II: Eigensystems, SIAM, Philadelphia, 1998.

[12] R. Vandebril, E. Van Camp, M. Van Barel, N. Mastronardi, Orthogonal similarity transformation of a symmetric matrix into a diagonal-plus-semiseparable one with free choice of the diagonal, Numerische Mathematik 102 (2006) 709–726.

[13] Y. Ikebe, On inverses of Hessenberg matrices, Linear Algebra and its Applications 24 (1979) 93–97.

[14] V.Y. Pan, X. Yan, Additive preconditioning, eigenspaces, and the inverse iteration, Technical Report TR 2007004, CUNY Ph.D. Program in Computer Science, Graduate Center, City University of New York, March 2007.

[15] V.Y. Pan, X. Yan, Null space and eigenspace computation with additive preconditioning, in: Jan Verschelde, Stephen Watt (Eds.), Proceedings of the Third International Workshop on Symbolic–Numeric Computation, SNC 2007, 152–160, July 2007, London, Ontario, Canada, ACM Press, New York, 2007.

[16] V.Y. Pan, D. Ivolgin, B. Murphy, R.E. Rosholt, I. Taj-Eddin, Y. Tang, X. Yan, Additive preconditioning and aggregation in matrix computations, Computers and Mathematics (with Applications), in press (doi:10.1016/j.camwa.2004.03.020).