2004

TR-2004009: A Reduction of the Matrix Eigenproblem to Polynomial Rootfinding via Similarity Transforms into Arrow-Head Matrices

Victor Y. Pan

A reduction of the matrix eigenproblem to polynomial rootfinding via similarity transforms into arrow-head matrices

Victor Y. Pan [*†]
Department of Mathematics and Computer Science
Lehman College of the City University of New York
Bronx, NY 10468, USA
vpan@lehman.cuny.edu

Abstract

We modify the customary approach to solving the algebraic eigenproblem. Instead of applying the QR algorithm to a Hessenberg matrix, we begin with the recent unitary similarity transform into a triangular plus rank-one matrix. Our novelty is nonunitary transforms of this matrix into similar arrow-head matrices, which we perform at a low arithmetic cost. The resulting eigenproblem can be effectively solved by the known algorithms. We also outline some directions for further work.

Key words: The algebraic eigenproblem, Similarity transforms, Arrow-head matrices.

[*]Supported by NSF Grant CCR 9732206 and PSC CUNY Award 65393-034.

[†]The results of this paper have been presented at the Workshop on Non-linear Approximation in Numerical Analysis, Moscow, Russia, June 2003, and at the SIAM Conference on Linear Algebra, Williamsburg, Virginia, July 2003.

1 Introduction

1.1 Our problem and the background

Our goal is the accelerated approximation of the eigensystem of a matrix, that is, its eigenvalues and eigenvectors. This is one of the two central problems of numerical linear algebra.

The customary solution for an $n \times n$ matrix M begins with its unitary similarity transform into a Hessenberg matrix H. The transform uses $(4/3)n^3 + O(n^2)$ arithmetic operations [GL96], [S98]. Hereafter we refer to arithmetic operations as ops. The method of choice at the next, final step is the celebrated QR algorithm. At this step the eigensystems of the matrices M and H are approximated. Statistically, the algorithm uses $10n^3$ ops on the average Hessenberg input [GL96, Section 7.5.6], but no upper bound is known for the worst case arithmetic complexity of this stage. Our present goal is to replace the QR stage by a distinct algorithm using $2n^3 + O(n^2)$ ops in the worst case.

We first replace the Hessenberg reduction step by a unitary similarity transform of an $n \times n$ matrix M into a TPR1 (that is, triangular plus rank-one) matrix $R + \mathbf{u}\mathbf{v}^T$ by using still $(4/3)n^3 + O(n^2)$ ops [VVMa]. Here the matrix R is upper triangular, $\mathbf{u} = (u_i)_{i=1}^n$ and $\mathbf{v} = (v_i)_{i=1}^n$ denote two column vectors, and $\mathbf{v}^T$ is the transpose of the vector $\mathbf{v}$, so that $\mathbf{u}\mathbf{v}^T$ is a rank-one matrix [I79].

Actually, the classes of Hessenberg and TPR1 matrices are closely related to one another; in particular the inverse of a non-singular unreduced Hessenberg matrix is a TPR1 matrix. For some matrices M, both transforms into Hessenberg and TPR1 matrices encounter reduced matrices, and then deflation reduces the size of the eigenproblem. Numerically, in both cases the deflation stage requires additional care; in the TPR1 case the problem has not been studied yet.

1.2 Our algorithms

In our present paper, we skip this study and assume that a TPR1 matrix is our starting point. Our main result is in devising two dual algorithms which transform any TPR1 matrix into similar arrow-head matrices of the forms $D + \mathbf{e}_1\mathbf{s}^T + \mathbf{t}\mathbf{e}_1^T$ and $D + \mathbf{e}_n\mathbf{s}^T + \mathbf{t}\mathbf{e}_n^T$, respectively. Here $D = \text{diag}(d_i)_{i=1}^n$ is a diagonal matrix and $\mathbf{e}_1 = (1, 0, \dots, 0)^T$ and $\mathbf{e}_n = (0, 0, \dots, 1)^T$ are two

coordinate vectors. The algorithms only require the solution of $n-1$ linear systems, each of $n-2$ equations, and $O(n^2)$ additional ops. Moreover, all these linear systems are triangular and diagonally dominant. This means the overall arithmetic cost of $2n^3 + O(n^2)$.

For the resulting arrow-head matrix, we have a number of effective eigen-solvers. Theoretically, we may approximate the eigenvalues of such a matrix as the roots of its characteristic polynomial, whose coefficients are readily available for a given arrow-head matrix. The known algorithms (see [P95], [P02], and the bibliography therein) compute all roots of a polynomial in linear arithmetic time up to a polylogarithmic factor. Furthermore, the transition from an eigenvalue to the associated eigenvectors takes linear time for an arrow-head matrix. Thus the overall number of arithmetic operations in our eigen-solver is still $2n^3 + O(n^2)$.

Practically, we may effectively approximate the eigensystem of an arrow-head matrix by applying the intensively tested algorithms in [F01], [BGP04], [BGPa]. [F01] combines the QR algorithm with the Weierstrass (Durand-Kerner) polynomial root-finder. The latter two papers adapt the inverse power and QR algorithms, respectively, to an arrow-head input matrix, whose structure is exploited to perform the algorithms in linear space by using linear time in every iteration steps. Furthermore, the well known robustness and rapid convergence of the algorithms are preserved in these adaptations according to the results of extensive numerical tests reported in [BGP04], [BGPa]. Thus we arrive at the desired eigen-solvers which use fewer ops.

1.3 Some alternative directions

DPR1, that is, diagonal plus rank-one, matrices are an alternative to arrow-head matrices. These two classes of matrices are closely related to one another (see our Remark 7.4) and can be solved equally efficiently by the cited algorithms. Unless R is a defective matrix (whose eigenspaces of the right and the left eigenvectors have a smaller dimension), a similarity transform from a TPR1 matrix $R + \mathbf{u}\mathbf{v}^T$ into a DPR1 matrix can be defined as follows:

$$R + \mathbf{u}\mathbf{v}^T \longrightarrow V^{-1}(R + \mathbf{u}\mathbf{v}^T)V = D + \mathbf{s}\mathbf{t}^T. \tag{1.1}$$

Here $D = V^{-1}RV$, $\mathbf{s} = V^{-1}\mathbf{u}$, $\mathbf{t}^T = \mathbf{v}^T V$, and V is a matrix of right eigenvectors of the matrix R. Unlike our transforms, however, this algorithm fails where R is a defective matrix and runs into problems where R is nearly defective.

Involving nonunitary similarity transforms, at the stage of the transition from TPR1 to arrow-head matrices, is a disadvantage for numerical implementation of our algorithms versus the QR algorithm. It is known that nonunitary transforms should be used with caution [GL96, Sections 7.1.5, 7.3.4 and 7.4.7]. Can we devise effective unitary similarity transforms instead? This task seems to be nontrivial. For instance, unitary similarity transform from the class of TPR1 matrices to arrow-head or DPR1 matrices is impossible if we require that the diagonal entries of the output matrix be real. Indeed, the classes of Hermitian matrices as well as of rank k matrices for any fixed k are closed under the unitary similarity transforms. The DPR1 and arrow-head matrices with real diagonals are subclasses of the HPR1 and HPR2 (that is, Hermitian plus rank-one and rank-two) matrices, respectively, and thus remain to be HPR1 and HPR2 matrices in unitary similarity transforms. Clearly, these classes do not cover the class of TPR1 matrices. In Section 8 we show simple unitary similarity transforms of an HPR1 matrix into a DPR1 matrix.

1.4 Organization of our paper

We organize the paper as follows. After some definitions in the next section, we specify Gauss similarity transforms in the general form in Section 3 and in a special case in Section 4. By recursively applying these transforms, we cancel most of the off-diagonal entries of a TPR1 matrix to arrive at the arrow-head matrices in Sections 5 and 6. In Section 7 we re-examine the transforms of a TPR1 matrix into a DPR1 matrix and represent these transforms as the sequences of Gauss transforms to compare with our algorithms in Sections 5 and 6; we also show the similarity of the non-defective DPR1 matrices to arrow-head matrices and of the unreduced arrow-head matrices to DPR1 matrices. In Section 8 we briefly discuss some further research directions.

Acknowledgements. I thank Luca Gemignani and Marc Van Barel for preprints and helpful discussions on TPR1 matrices.

2 Definitions

$\mathbb{R}^{n\times n}$ is the algebra of real $n\times n$ matrices. $M=(m_{i,j})_{i,j=1}^{n}$ is an $n\times n$ matrix, $\mathbf{v}=(v_i)_{i=1}^{n}$ is a column vector of dimension n, and M^T and $\mathbf{v}^T$ are their

transposes. I_k is the $k\times k$ identity matrix, $I = I_n$; $\mathbf{e}_j$ is the j-th column vector of I, so that $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$, $\mathbf{e}_n = (0, 0, \ldots, 1)^T$. $\mathbf{0}$ is the null vector of a fixed dimension. $\mathbf{1} = (1, \ldots, 1)^T$. $R = (r_{i,j})_{i,j=1}^n$ is an $n \times n$ upper triangular matrix with $r_{i,j} = 0$ for $i > j$. $D =\mathrm{diag}(d_i)_{i=1}^n$ is the $n \times n$ diagonal matrix with the diagonal entries $d_1, \ldots, d_n$. *DPR1* and *TPR1* denote the two classes of $n \times n$ matrices of the form $D + \mathbf{u}\mathbf{v}^T$ and $R + \mathbf{u}\mathbf{v}^T$, respectively. The NW and SE *arrow-head matrices* are the two classes of $n \times n$ matrices of the form $D+\mathbf{e}_h\mathbf{s}^T+\mathbf{t}\mathbf{e}_h^T$ for $h = 1$ and $h = n$, respectively. $C^{(k)} = (c_{i,j}^{(k)})$ for $k = 2, \ldots, n$ is the $k\times k$ matrix of cyclic permutation such that $c_{i,j}^{(k)} = 1$ if $(i-j) \bmod k = 1$, $c_{i,j}^{(k)} = 0$ otherwise. (We have $C^{(k)}\mathbf{v} = (v_k, v_1, \ldots, v_{k-1})^T$ for $\mathbf{v} = (v_j)_{j=1}^k$ and $C^{(k)T}C^{(k)} = I_k$.) $P_k =\mathrm{diag}(C^{(k)}, I_{n-k})$, $P^{(k)} =\mathrm{diag}(I_{n-k}, C^{(k)})$. (We have $P_k^T P_k = P^{(k)T}P^{(k)} = I$, and the similarity transforms $M \longrightarrow P_k^T M P_k$ and $M \longrightarrow P^{(k)}MP^{(k)T}$ cyclically interchange the first k and the last k rows and columns of an $n \times n$ matrix M, respectively.)

3 Gauss Similarity Transforms

Our basic tool is the Gauss similarity transforms (of the first kind)

$G_{\mathbf{x}} : \mathbb{R}^{n\times n} \longrightarrow \mathbb{R}^{n\times n}$,

$G_{\mathbf{x}}(M) = (I + \mathbf{e}_1\mathbf{x}^T)M(I - \mathbf{e}_1\mathbf{x}^T)$,

defined by vectors $\mathbf{x} = (x_j)_{j=1}^n$ where $x_1 = 0$. We immediately observe the following property.

Fact 3.1. *For $x_1 = 0$, the Gauss transform $G_{\mathbf{x}}$ is a similarity transform, that is, $(I + \mathbf{e}_1\mathbf{x}^T)(I - \mathbf{e}_1\mathbf{x}^T) = I$.*

Hereafter we call $G_{\mathbf{x}}$ the Gauss similarity transform (of the first kind).

Theorem 3.2. *The Gauss similarity transform $G_{\mathbf{x}}$ maps a TPR1 matrix $T = R + \mathbf{u}\mathbf{v}^T$ into a TPR1 matrix. More precisely,*

$G_{\mathbf{x}}(T) = G_{\mathbf{x}}(R) + G_{\mathbf{x}}(\mathbf{u}\mathbf{v}^T)$,

$G_{\mathbf{x}}(R) = R + \mathbf{e}_1\mathbf{x}^T(R - r_{1,1}I)$,

$G_{\mathbf{x}}(\mathbf{u}\mathbf{v}^T) = (\mathbf{u} + \mathbf{e}_1(\mathbf{x}^T\mathbf{u}))(\mathbf{v}^T - v_1\mathbf{x}^T)$.

By setting $a = u_1 + \mathbf{x}^T\mathbf{u}$, we obtain that the first row of the matrix $G_{\mathbf{x}}(T)$ equals $\mathbf{w}^{(1)T} = \mathbf{e}_1^T G_{\mathbf{x}}(T) = \mathbf{e}_1^T R + \mathbf{x}^T(R - r_{1,1}I) + a(\mathbf{v}^T - v_1\mathbf{x}^T) = \mathbf{e}_1^T R + a\mathbf{v}^T + \mathbf{x}^T(R - (r_{1,1} + av_1)I)$, whereas the i-th row for $i > 1$ equals $\mathbf{w}^{(i)T} = \mathbf{e}_i^T G_{\mathbf{x}}(T) = \mathbf{e}_i^T R + u_i(\mathbf{v}^T - v_1\mathbf{x}^T)$. In particular for $u_i = 0$ and $i > 0$, the i-th rows of the matrices $G_{\mathbf{x}}(T)$ and T coincide with one another.

Proof. We have the following equations, $G_{\mathbf{x}}(R) = (I + \mathbf{e}_1\mathbf{x}^T)R(I - \mathbf{e}_1\mathbf{x}^T) = (R + \mathbf{e}_1\mathbf{x}^T R)(I - \mathbf{e}_1\mathbf{x}^T) = R - R\mathbf{e}_1\mathbf{x}^T + \mathbf{e}_1\mathbf{x}^T R - \mathbf{e}_1\mathbf{x}^T R\mathbf{e}_1\mathbf{x}^T$. Substitute the equations $R\mathbf{e}_1 = r_{1,1}\mathbf{e}_1$ and $\mathbf{x}^T R\mathbf{e}_1 = r_{1,1}\mathbf{x}^T\mathbf{e}_1 = r_{1,1}x_1 = 0$ and obtain that $G_{\mathbf{x}}(R) = R + \mathbf{e}_1\mathbf{x}^T(R - r_{1,1}I)$. Combine this equation with the following one, $G_{\mathbf{x}}(\mathbf{u}\mathbf{v}^T) = (I + \mathbf{e}_1\mathbf{x}^T)\mathbf{u}\mathbf{v}^T(I - \mathbf{e}_1\mathbf{x}^T) = (\mathbf{u} + \mathbf{e}_1(\mathbf{x}^T\mathbf{u}))(\mathbf{v}^T - v_1\mathbf{x}^T)$ and deduce the theorem. □

In virtue of Theorem 3.2, Gauss similarity transform maps a TPR1 matrix into a TPR1 matrix. Furthermore, we may represent both matrices in the same form $R + \mathbf{u}\mathbf{v}^T$ where only the vector $\mathbf{v}$ and the first row $\mathbf{e}_1^T R$ of the matrix R change. The changes in the vector $\mathbf{e}_1^T R$ are specified and analyzed in Theorem 4.1 in the next section. The other changes can be computed in $2n - 2$ ops.

4 Gauss Row Cancelation Transforms

Let us specialize a Gauss transform of a TPR1 matrix $T = R + \mathbf{u}\mathbf{v}^T$ to cancel all the non-corner entries in the first row. In the next section such *Gauss row cancelation transforms* recursively transform a TPR1 matrix into an SE arrow-head matrix.

Denote by $\tilde{\mathbf{x}}$, $\tilde{\mathbf{w}}$, and $\tilde{\mathbf{r}}$ the subvectors of the dimension $n-2$ obtained by deleting the pairs of the first and the last components of the vectors $\mathbf{x}$, $\mathbf{w}^{(1)} = \mathbf{e}_1^T G_{\mathbf{x}}(T)$, and $\mathbf{e}_1^T R + a\mathbf{v}^T$, respectively. Denote by $R(a)$ the $(n-2)\times(n-2)$ submatrix obtained by deleting the pairs of the first and the last rows and the first and the last columns of the matrix $R - (r_{1,1} + av_1)I$. In virtue of Theorem 3.2, we have

$$\tilde{\mathbf{w}}^T = \tilde{\mathbf{r}}^T + \tilde{\mathbf{x}}^T R(a). \tag{4.1}$$

Assumption 1. Hereafter until Section 7 and in Remark 7.4, we assume that $v_1 u_n \neq 0$ for all TPR1 matrices of the form $R + \mathbf{u}\mathbf{v}^T$ that we encounter. (If this were not the case, we could have just deflated the TPR1 matrices.)

Theorem 4.1. (A) *Given a TPR1 matrix $R + \mathbf{u}\mathbf{v}^T$ such that $v_1 u_n \neq 0$, we can compute a scalar a and a vector $\mathbf{x}$ such that $G_{\mathbf{x}}$ is a Gauss row cancelation transform for this matrix, that is, $\tilde{\mathbf{w}} = \mathbf{0}$ for the vector $\tilde{\mathbf{w}}^T$ in (4.1).*
(B) *Moreover, this computation involves selection of a (large) scalar a, solution of a triangular and diagonally dominant linear system of $n-2$ equations with the coefficient matrix R_a, and besides that performing $O(n)$ ops.*

Proof. Theorem 4.1 is supported by the following procedure.

Procedure 1. Row cancelation.
1. Fix a scalar a for which the matrix $R(a)$ is diagonally dominant (recall that $v_1 \neq 0$). (Formally it is sufficient to have a non-singular matrix $R(a)$, but its diagonal dominance implies better numerical stability of the computations at the next stage.)
2. Compute the vector $\tilde{\mathbf{x}}^T = -\tilde{\mathbf{r}}R^{-1}(a)$.
3. Compute the scalar x_n such that $u_1 + \mathbf{x}^T\mathbf{u} = a$ for $\mathbf{x}^T = (0, \tilde{\mathbf{x}}^T, x_n)^T$ (recall that $u_n \neq 0$) and output the vector $\mathbf{x}$.

Since we let $a = u_1 + \mathbf{x}^T\mathbf{u}$ at Stage 3, we deduce equation (4.1) from Theorem 3.2. Substitute this equation in Stage 2 to prove Part(A). Part(B) is immediately verified. □

5 Transformation into an SE Arrow-head Matrix

The next definition and theorem define the k-th step of our main algorithm (see Algorithm 5.3 and Figure 1).

Definition 5.1. A TPR1 matrix T_k is a k-th *level approximant to an SE (resp. NW)* arrow-head matrix if all its off-diagonal entries in the k last (resp. first) rows and columns equal zero, except possibly for the entries in the last (resp. first) row and column. In particular, an $n \times n$ arrow-head matrix is an n-th level approximant to itself, whereas a TPR1 matrix is a first level approximant to SE and NW arrow-head matrices.

Theorem 5.2. *Let an $n \times n$ TPR1 matrix T_k be a k-th level approximant to an SE arrow-head matrix for some k in the range from 1 to $n-1$. Let the matrix $G_{\mathbf{x}_k}(T_k) = (I + \mathbf{e}_1\mathbf{x}_k^T)T_k(I - \mathbf{e}_1\mathbf{x}_k^T)$ be output by the Gauss row cancellation transform for the matrix T_k. Let P_{n-k} denote the permutation matrix defined in Section 2. Then $T_{k+1} = P_{n-k}^T G_{\mathbf{x}_k}(T_k)P_{n-k}$ for $k < n-1$ or $T_n = G_{\mathbf{x}_k}(T_k)$ for $k = n-1$ is a TPR1 matrix which is a $(k+1)$-st level approximant to an SE arrow-head matrix.*

Proof. By the definition of the Gauss row cancelation transforms, the non-corner entries in the first row of the matrix $G_{\mathbf{x}_k}(T_k) = R_k + \mathbf{u}_k\mathbf{v}_k^T$ are zeros for

all k. Here R_k is an upper triangular matrix, and with no loss of generality we assume that the first coordinate of the vector $\mathbf{u}_k$ vanishes. The cyclic similarity permutation $G_{\mathbf{x}_k}(T_k) \longrightarrow P_{n-k}^T\ G_{\mathbf{x}_k}(T_k)\ P_{n-k}$ transforms the matrix R_k into an upper triangular matrix with the $(n-k)$-th row filled with zeros (except possibly for the two entries in the last column and on the diagonal) and transforms the vector $\mathbf{u}_k$ into the vector $P_{n-k}^T\mathbf{u}_k$ with the vanishing $(n-k)$-th coordinate. Because of the latter property and Theorem 3.2, our subsequent applications of the Gauss transforms keep the $(n-k)$-th rows of the matrices T_{k+i} and the $(n-k)$-th coordinates of the vectors $\mathbf{u}_{k+i}$ unchanged for all $i > 0$ (note that the cyclic permutations $P_{n-k-i}^T G_{\mathbf{x}_{k+i}}(T_{k+i})P_{n-k-i}$ do not affect these row and entry). Therefore, the $n-2$ zero entries in the $(n-k)$-th row of the matrix T_{k+1} are not changed in the transition to the matrices T_{k+1+i} for $i = 1, 2, \ldots$. This completes the proof. □

We are ready to specify our main algorithm (see Figure 1).

Algorithm 5.3. Similarity transform into an SE arrow-head matrix.
INPUT: an $n \times n$ TPR1 matrix $T_1 = R + \mathbf{u}\mathbf{v}^T$.
OUTPUT: an SE arrow-head matrix T_n and a sequence of vectors $\mathbf{x}_k$ of dimension n for $k = 1, \ldots, n-1$ such that $T_n = S^{-1}T_1S$, $S = \prod_{k=1}^{n-2}((I - \mathbf{e}_1\mathbf{x}_k^T)P_{n-k})(I - \mathbf{e}_1\mathbf{x}_{n-1}^T)$, and $S^{-1} = (I + \mathbf{e}_1\mathbf{x}_{n-1}^T)\prod_{k=2}^{n-1}(P_k^T(I + \mathbf{e}_1\mathbf{x}_{n-k}^T))$.
COMPUTATIONS: recursively compute the vectors $\mathbf{x}_k$, $k = 1, \ldots, n-1$, and the TPR1 matrices $T_{k+1} = P_{n-k}^T G_{\mathbf{x}_k}(T_k)P_{n-k}$, $k = 1, \ldots, n-2$; $T_n = G_{\mathbf{x}_{n-1}}(T_{n-1})$, similar to the input matrix T_1; here $G_{\mathbf{x}_k}(T_k) = (I + \mathbf{e}_1\mathbf{x}_k^T)T_k(I - \mathbf{e}_1\mathbf{x}_k^T)$ for $k = 1, \ldots, n-1$ are the Gauss row cancelation transforms for the matrices T_k and some selected scalars a_k.

Due to Theorem 5.2, the matrices T_{k+1} for $k = 0, 1, \ldots, n-1$ are the $(k+1)$st level approximants to an SE arrow-head matrix. In particular T_n is an SE arrow-head matrix. This shows correctness of the algorithm.

Besides the row and column interchange, the computations in Algorithm 5.3 amount to the application of $n-1$ Gauss cancelation transforms. In virtue of Theorem 4.1(B), this means the solution of $n-1$ triangular and diagonally dominant linear systems (each of $n-2$ equations) and, in addition, performing $O(n^2)$ ops.

6 A Dual Transformation into an NW Arrowhead Matrix

Numerical stability problems in Algorithm 5.3 grow with $||S||_2\ ||S^{-1}||_2$, the condition number of the matrix S. We have $||S||_2 \le \prod_{k=1}^{n-1} ||I - \mathbf{e}_1\mathbf{x}_k^T||_2 \le \prod_{k=1}^{n-1}(1 + ||\mathbf{x}_k||_2)$, $||S^{-1}||_2 \le \prod_{k=1}^{n-1} ||I + \mathbf{e}_1\mathbf{x}_k^T||_2 \le \prod_{k=1}^{n-1}(1 + ||\mathbf{x}_k||_2)$. To bound the latter product, we may choose the $n-1$ parameters $a = a_k$ for $n-1$ vectors $\mathbf{x}_k, k = 1, \dots, n-1$, and we may modify TPR1 representation of the matrix T_1, but the overall impact of all this is not easy to control. Next we present a dual version of Algorithm 5.3 to double our chances for yielding a numerically stable solution algorithm. This version is obtained by interchanging the roles of the vectors $\mathbf{u}$ and $\mathbf{v}$ as well as the last column/row and the first row/column of the matrix R. Furthermore we change the Gauss transforms of the first kind studied in the previous sections into the Gauss transforms of the second kind defined as follows:

$G^{(\mathbf{y})} : \mathbb{R}^{n\times n} \longrightarrow \mathbb{R}^{n\times n}$,

$M \longrightarrow G^{(\mathbf{y})}(M) = (I + \mathbf{y}\mathbf{e}_n^T)M(I - \mathbf{y}\mathbf{e}_n^T)$

where $\mathbf{y} = (y_i)_{i=1}^n$, $y_n = 0$.

Let us extend Fact 3.1 and Theorem 3.2.

Fact 6.1. *For $y_n = 0$, the Gauss transform $G^{(\mathbf{y})}$ of the second kind is a similarity transform, that is,*

$$(I + \mathbf{y}\mathbf{e}_n^T)(I - \mathbf{y}\mathbf{e}_n^T) = I.$$

Proof. By inspection. □

Hereafter we call $G^{(\mathbf{y})}$ the Gauss similarity transform of the second kind.

Theorem 6.2. *The Gauss similarity transform of the second kind $G^{(\mathbf{y})}$ transforms a TPR1 matrix $T = R + \mathbf{u}\mathbf{v}^T$ into a TPR1 matrix. More precisely,*

$G^{(\mathbf{y})}(T) = G^{(\mathbf{y})}(R) + G^{(\mathbf{y})}(\mathbf{u}\mathbf{v}^T)$,

$G^{(\mathbf{y})}(R) = R - (R - r_{n,n}I)\mathbf{y}\mathbf{e}_n^T$,

$G^{(\mathbf{y})}(\mathbf{u}\mathbf{v}^T) = (\mathbf{u} + u_n\mathbf{y})(\mathbf{v}^T - (\mathbf{v}^T\mathbf{y})\mathbf{e}_n^T)$.

By setting $b = v_n - \mathbf{v}^T\mathbf{y}$, we obtain that the last column of the matrix $G^{(\mathbf{y})}(T)$ equals $G^{(\mathbf{y})}(T)\mathbf{e}_n = R\mathbf{e}_n + b\mathbf{u} - (R - (r_{n,n} + bu_n)I)\mathbf{y}$, whereas the j-th column for $i < n$ equals $G^{(\mathbf{y})}(T)\mathbf{e}_j = R\mathbf{e}_j + v_j(\mathbf{u} + u_n\mathbf{y})$. In particular for $v_j = 0$ and $j < n$, the j-th columns of the matrices $G^{(\mathbf{y})}(T)$ and T coincide with one another.

Proof. We have the following equations, $G^{(\mathbf{y})}(R) = (I + \mathbf{y}\mathbf{e}_n^T)R(I - \mathbf{y}\mathbf{e}_n^T) = (R + \mathbf{y}\mathbf{e}_n^T R)(I - \mathbf{y}\mathbf{e}_n^T) = R + \mathbf{y}\mathbf{e}_n^T R - R\mathbf{y}\mathbf{e}_n^T - \mathbf{y}\mathbf{e}_n^T R\mathbf{y}\mathbf{e}_n^T$. Substitute the equations $\mathbf{e}_n^T R = r_{n,n}\mathbf{e}_n^T$ and $\mathbf{e}_n^T R\mathbf{y} = r_{n,n}\mathbf{e}_n^T\mathbf{y} = r_{n,n}y_n = 0$ and obtain that $G^{(\mathbf{y})}(R) = R + (r_{n,n}I - R)\mathbf{y}\mathbf{e}_n^T$. Combine this equation with the following one, $G^{(\mathbf{y})}(\mathbf{u}\mathbf{v}^T) = (I + \mathbf{y}\mathbf{e}_n^T)\mathbf{u}\mathbf{v}^T(I - \mathbf{y}\mathbf{e}_n^T) = (\mathbf{u} + u_n\mathbf{y})(\mathbf{v}^T - (\mathbf{v}^T\mathbf{y})\mathbf{e}_n^T)$ and deduce the theorem. □

In virtue of Theorem 6.2, a Gauss similarity transform of the second kind maps a TPR1 matrix into a TPR1 matrix. Furthermore, we may represent both matrices in the same form $R + \mathbf{u}\mathbf{v}^T$ where only the vector $\mathbf{v}$ and the last column $R\mathbf{e}_n$ of the matrix R change. The changes in the vector $R\mathbf{e}_n$ are studied next. The other change can be computed in $2n - 2$ ops.

Write R_b to denote the $(n-2) \times (n-2)$ submatrix obtained by deleting the pair of the first and last rows and the pair of the first and last columns of the matrix $R - (r_{n,n} + bu_n)I$.

Theorem 6.3. (A) *Given a TPR1 matrix $T = R + \mathbf{u}\mathbf{v}^T$ such that $u_1 v_n \neq 0$, we compute a scalar b and a vector $\mathbf{y}$ such that $G^{(\mathbf{y})}$ is the Gauss column cancellation transform for the matrix T, that is, the last column of the matrix $G^{(\mathbf{y})}(T)$ vanishes, possibly except for its two corner entries.*

(B) *Moreover, this computation involves selection of a (large) scalar b, solution of a triangular and diagonally dominant linear system of $n-2$ equations with the coefficient matrix R_b, and besides that $O(n)$ ops.*

Proof. Theorem 6.3 is proved similarly to Theorem 4.1. We omit some simple details but specify the supporting procedure dual to Procedure 1. □

Procedure 2. Column cancelation.
1. Fix a scalar b for which the matrix R_b is diagonally dominant (recall that $u_n \neq 0$).
2. Compute the components $y_2, \dots, y_{n-1}$ of the vector $\mathbf{y}$ by solving a triangular linear system of $n-2$ equations with the coefficient matrix R_b.
3. Recall that $y_n = 0$, $v_1 \neq 0$ and compute the remaining component y_1 of the vector $\mathbf{y}$ by solving the equation $v_n - \mathbf{v}^T\mathbf{y} = b$.

Let us specify the dual extension of Theorem 6.2 and Algorithm 5.3 (see Figure 2).

Theorem 6.4. *Let an $n \times n$ TPR1 matrix $T^{(k)}$ be a k-th level approximant to an NW arrow-head matrix for some k in the range from 1 to $n-1$. Let the matrix $G^{(\mathbf{y}^{(k)})}(T^{(k)}) = (I+\mathbf{y}^{(k)}\mathbf{e}_n^T)T^{(k)}(I-\mathbf{y}^{(k)}\mathbf{e}_n^T)$ be output by the Gauss column cancellation transform for the matrix $T^{(k)}$. Let $P^{(n-k)}$ denote the permutation matrix defined in Section 2. Then $T^{(k+1)} = P^{(n-k)}G^{(\mathbf{y}^{(k)})}(T^{(k)})P^{(n-k)T}$ for $k < n-1$ or $T^{(n)} = G^{(\mathbf{y}^{(k)})}(T_k)$ for $k = n-1$ is a TPR1 matrix which is a $(k+1)$-st level approximant to an NE arrow-head matrix.*

Proof. The theorem is proved similarly to Theorem 5.2. □

Algorithm 6.5. Similarity transform into an NW arrow-head matrix.
INPUT: an $n \times n$ TPR1 matrix $T^{(1)} = R + \mathbf{u}\mathbf{v}^T$.
OUTPUT: an NW arrow-head matrix $T^{(n)}$ and a sequence of vectors $\mathbf{y}^{(k)}$ of dimension n for $k = 1, \dots, n-1$ such that $T^{(n)} = S^{-1}T^{(1)}S$, $S = \prod_{k=1}^{n-2}((I - \mathbf{y}^{(k)}\mathbf{e}_n^T)P^{(n-k)T})(I - \mathbf{y}^{(n-1)}\mathbf{e}_n^T)$, and $S^{-1} = (I + \mathbf{y}^{(n-1)}\mathbf{e}_n^T)\prod_{k=2}^{n-1}(P^{(k)}(I + \mathbf{y}^{(n-k)}\mathbf{e}_n^T))$.
COMPUTATIONS: recursively compute the vectors $\mathbf{y}^{(k)}$ and the TPR1 matrices $T^{(k+1)} = P^{(n-k)}G^{(\mathbf{y}^{(k)})}(T^{(k)})P^{(n-k)T}$ for $k = 1, \dots, n-2$, $T^{(n)} = G^{(\mathbf{y}^{(n)})}(T^{(n-1)})$, which are similar to the input matrix $T^{(1)}$; here $G^{(\mathbf{y}^{(k)})}(T^{(k)}) = (I + \mathbf{y}^{(k)}\mathbf{e}_n^T)T^{(k)}(I - \mathbf{y}^{(k)}\mathbf{e}_n^T)$ for $k = 1, \dots, n-1$ are the Gauss column cancelation transforms for the matrices $T^{(k)}$ and the selected scalars b_k.

The algorithm performs similarly to Algorithm 5.3 and uses as many ops but eliminates the matrix entries in the reverse order: the off-diagonal entries of the matrix $T^{(k)}$ for $k > 1$ are set to zero (and remain equal to 0 in the matrices $T^{(k+i)}$ for $i > 0$) if they lie in the k-th column but not in the first row.

7 Alternative Transforms with Diagonalization

An alternative similarity transform of a TPR1 matrix into a DPR1 matrix is possible if the matrix R is non-defective, that is, has non-singular matrices S and S^{-1} of its right and left eigenvectors, respectively. This holds, e.g., if all diagonal entries of the matrix R are distinct. Then we have $S^{-1}RS = D = \text{diag}(r_{i,i})_{i=1}^n$, and therefore $S^{-1}(R + \mathbf{u}\mathbf{v}^T)S = D + \mathbf{s}\mathbf{t}^T$ where $\mathbf{s} = S^{-1}\mathbf{u}$, $\mathbf{t}^T = \mathbf{v}^T S$. This gives us a similarity transform of a TPR1 matrix $R + \mathbf{u}\mathbf{v}^T$

(with distinct diagonal entries of R) into a DPR1 matrix. The matrices S and S^{-1} can be computed based on $2n$ applications of the shifted inverse power method whose each step amounts to solving a triangular linear system of n equations with the matrices $R - r_{i,i}I$ for $1 \le i \le n$.

Alternatively, we may diagonalize the matrix R based on recursive application of the Gauss transforms $G_{\mathbf{x}}$ and $G^{(\mathbf{y})}$. Let us show some details of this approach (see Figures 3 and 4) and compare it with Algorithms 5.3 and 6.5 applied in the case where $\mathbf{u} = \mathbf{v} = \mathbf{0}$. (This means relaxing Assumption 1.) The comparison should enable a better insight into both algorithms and the deflation technique. The identities $\mathbf{u} = \mathbf{v} = \mathbf{0}$ imply that $b = v_n - \mathbf{v}^T\mathbf{y} = a = u_1 + \mathbf{x}^T\mathbf{u} = \mathbf{0}$ identically in $\mathbf{x}$ and $\mathbf{y}$, which leaves us with the two additional parameters x_n and y_1 and also simplifies Theorems 3.2 and 6.2 as follows.

Theorem 7.1. *For an $n \times n$ upper triangular matrix R and two vectors $\mathbf{x} = (x_i)_{i=1}^n$ and $\mathbf{y} = (y_i)_{i=1}^n$ where $x_1 = y_n = 0$, the matrices $G_{\mathbf{x}}(R)$ and $G^{(\mathbf{y})}(R)$ are upper triangular; furthermore we have the following expressions for the rows of the matrix $G_{\mathbf{x}}(R)$ and the columns of the matrix $G^{(\mathbf{y})}(R)$:*
$\mathbf{e}_1^T G_{\mathbf{x}}(R) = \mathbf{e}_1^T R + \mathbf{x}^T(R - r_{1,1}I)$,
$\mathbf{e}_i^T G_{\mathbf{x}}(R) = \mathbf{e}_i^T R$ *for* $i > 1$,
$G^{(\mathbf{y})}(R)\mathbf{e}_n = R\mathbf{e}_n - (R - r_{n,n}I)\mathbf{y}$,
$G^{(\mathbf{y})}(R)\mathbf{e}_j = R\mathbf{e}_j$ *for* $j < n$.

We cancel the same entries of the first row or the last column as in Sections 5 and 6 and in addition cancel the corner entry $(1, n)$ by using the extra parameters x_n or y_1. Due to Theorem 7.1, we achieve this cancelation by applying the Gauss transforms from the previous sections, but for $\mathbf{x}$ and $\mathbf{y}$ that satisfy the following equations:

$$(\mathbf{e}_1^T R + \mathbf{x}^T(R - r_{1,1}I))P_{\leftarrow}^T = \mathbf{0}, \quad P_{\rightarrow}(R\mathbf{e}_n - (R - r_{n,n}I)\mathbf{y}) = \mathbf{0}. \tag{7.1}$$

Here $P_{\leftarrow} = [\mathbf{0}\ I_{n-1}]$ and $P_{\rightarrow} = [I_{n-1}\ \mathbf{0}]$ denote two projection matrices of the size $(n-1) \times n$, which shift a vector $\mathbf{v}$ of dimension n into its trailing and its leading subvectors of dimension $n-1$, respectively. In addition to choosing $x_1 = y_n = 0$, we define the other $n-1$ components of each of the vectors $\mathbf{x}$ and $\mathbf{y}$ by solving the above triangular linear systems, each of $n-1$ equations and each non-singular for R having n distinct diagonal entries. Similarly to Algorithms 5.3 and 6.5, we extend these Gauss cancelation transforms recursively, and in $n-1$ recursive steps yield the diagonalization. In Algorithms

7.2 and 7.3 below we outline these computations. The algorithms consists of recursive application of the Gauss (row and column) cancelation transforms to the matrices R_k and $R^{(k)}$, respectively, for $k = 1, \dots, n-1$ where R_1 or $R^{(1)}$ is the input matrix R. (See Figures 3 and 4). The Gauss transforms are defined by the vectors $\mathbf{x}_k$ and $\mathbf{y}^{(k)}$ which satisfy (7.1) for $R = R_k$ and $R = R^{(k)}$, respectively.

Algorithm 7.2. [Extending Algorithm 5.3.]
For a non-defective input matrix $R = R_1$, recursively compute the matrices $R_{k+1} = P_{n-k}^T G_{\mathbf{x}_k}(R_k) P_{n-k}$, $k = 1, \dots, n-2$, $R_n = G_{\mathbf{x}_{n-1}}(R_{n-1})$.
OUTPUT: the matrix $R_n = D$ and the vectors $\mathbf{x}_k$ for $k = 1, \dots, n-1$.

Algorithm 7.3. [Extending Algorithm 6.5.]
For a non-defective input matrix $R = R^{(1)}$, recursively compute the matrices $R^{(k+1)} = P^{(n-k)} G^{(\mathbf{y}^{(k)})}(R^{(k)}) P^{(n-k)T}$, $k = 1, \dots, n-2$, $R^{(n)} = G^{(\mathbf{y}^{(n-1)})}(R^{(n-1)})$.
OUTPUT: the matrix $R^{(n)} = D$ and the vectors $\mathbf{y}^{(k)}$ for $k = 1, \dots, n-1$.

It is easily verified that all the off-diagonal entries in the last k columns of the matrix R_k and in the last k rows of the matrix $R^{(k)}$ vanish for every k. In particular, $R_n = R^{(n)} = \mathrm{diag}(r_{i,i})_{i=1}^n$. Due to the row/column cancelation in the process of performing Algorithms 7.2 and 7.3, the size of triangular linear systems defining the vectors $\mathbf{x}_k$ and $\mathbf{y}^{(k)}$ (and consequently defining the Gauss cancelation transforms) decreases by one in each recursive step. This means saving about 2/3 of all ops versus Algorithms 5.3 and 6.5, but the price for this is the lack of the diagonal dominance in the triangular systems solved in Algorithms 7.2 and 7.3.

Remark 7.4. The output matrices of Algorithms 5.3 and 6.5 are arrow-head matrices versus the DPR1 matrices output by Algorithms 7.2 and 7.3. The two matrix classes, however, are close to one another. Indeed, an NW arrow-head matrix $A = D + \mathbf{e}_1 \mathbf{s}^T + \mathbf{t}\mathbf{e}_1^T$, $D = \mathrm{diag}(d_i)_i$, is a special case of a TPR1 matrix $R + \mathbf{u}\mathbf{v}^T$ for $R = D + \mathbf{e}_1 \mathbf{s}^T$, $\mathbf{u} = \mathbf{t}$, $\mathbf{v} = \mathbf{e}_1$. If this matrix R is non-defective, then its diagonalizations in this section preserve the rank one for the matrix $\mathbf{t}\mathbf{e}_1^T$, that is, transform the matrix A into a DPR1 matrix. Similar observations hold for the SE arrow-head matrices. Now, conversely, consider a DPR1 matrix $M = D + \mathbf{u}\mathbf{v}^T$ where $v_1 \neq 0$. (If $v_1 = 0$, we may deflate M.) Apply the Gauss transform $G_{\mathbf{x}}$ where $\mathbf{x} = (x_j)_{j=1}^n$, $x_1 = 0$, $x_j = v_j / v_1$ for $j = 2, \dots, n$. By Theorem 3.2 we have $G_{\mathbf{x}}(M) =$

$D+v_1(\mathbf{u}+(\mathbf{x}^T\mathbf{u})\mathbf{e}_1)\mathbf{e}_1^T+\mathbf{e}_1\mathbf{x}^T(D-d_1I)$, and so $G_{\mathbf{x}}(M)$ is an NW arrow-head matrix. Likewise, if $u_n\neq 0$ (otherwise we may deflate M), then we deduce from Theorem 6.2 that $G^{(\mathbf{y})}(M)=D+u_n\mathbf{e}_n(\mathbf{v}^T-(\mathbf{v}^T\mathbf{y})\mathbf{e}_n^T)-(D-d_nI)\mathbf{y}\mathbf{e}_n^T$ where $\mathbf{y}=(y_i)_{i=1}^n$, $y_n=0$, $y_i=-u_i/u_n$ for $i=1,\dots,n-1$, and so $G^{(\mathbf{y})}(M)$ is an SE arrow-head matrix.

Remark 7.5. For the characteristic polynomial $c_A(\lambda)$ of an NW arrow-head matrix $A=D+\mathbf{e}_1\mathbf{s}^T+\mathbf{t}\mathbf{e}_1^T$, we have

$$c_A(\lambda)=(\lambda-w_1-\sum_{k=2}^{n}w_k/(\lambda-d_k))\prod_{i=2}^{n}(\lambda-d_i). \tag{7.2}$$

Here $D=\mathrm{diag}(d_i)_{i=1}^n$, $w_1=d_1+s_1+t_1$; $w_k=s_kt_k$, $k=2,\dots,n$. The values $c_A(\lambda_i)$ for $i=1,\dots,n$ and any set $\lambda_1,\dots,\lambda_n$ can be computed in $5n^2+O(n)$ ops. Based on (7.2) we may readily compute an NW arrow-head matrix A such that $c_T(\lambda)=c(\lambda)$ for a given polynomial $c(\lambda)$. Similar properties hold for the SE arrow-head matrices.

8 Discussion

Our algorithms yield a novel solution of the algebraic eigenproblem at the record low arithmetic cost. Since they involve nonunitary similarity transforms, some further theoretical and experimental study is in order. As we observed in the introduction, the similarity transform of a general or even a general TPR1 matrix A into an HPR1 or HPR2 matrix cannot be unitary, whereas the customary unitary similarity transform maps an HPR1 matrix H into a tridiagonal Hermitian plus rank-one matrix. (HPRk for $k=1,2$ is our abbreviation for Hermitian plus rank-k.) At this point, we may apply the QR algorithm or other known effective algorithms to diagonalize the tridiagonal Hermitian term of this HPR1 matrix and thus to complete the mapping of the original HPR1 matrix into a DPR1 matrix with a real diagonal. This suggests the following objectives for our future work.

1. Among the non-unitary similarity transforms of a matrix A into a Hermitian plus rank-one matrix H, seek a reasonably fast ones which are most numerically stable. Our present algorithms can be viewed as

the initial step in this direction.

2. Devise unitary similarity transforms of a matrix A into non-Hermitian condensed matrices M (such as tridiagonal plus small rank matrices). In particular if the matrix $M - \lambda I$ is readily invertible (e.g., is a tridiagonal plus small rank matrix), then the inverse power iteration in [BGP04] is still effective.

References

[BGP04] 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**, 447–459, 2004.

[BGPa] D. A. Bini, L. Gemignani, V. Y. Pan, QR-like algorithms for generalized semiseparable matrices, Technical Report 1470, *Department of Math, University of Pisa*, Pisa, Italy, July 2003.

[F01] S. Fortune, Polynomial Root Finding Using Iterated Eigenvalue Computation, *Proc. International Symposium on Symbolic and Algebraic Computation (ISSAC'01)*, 121–128, ACM Press, New York, 2001.

[GL96] G. H. Golub, C. F. Van Loan, *Matrix Computations*, 3rd edition, The Johns Hopkins University Press, Baltimore, Maryland, (1996).

[I79] Y. Ikebe, On inverses of Hessenberg matrices, *Linear Algebra and Its Applications*, **24**, 93–97, 1979.

[P95] V. Y. Pan, Optimal (up to Polylog Factors) Sequential and Parallel Algorithms for Approximating Complex Polynomial Zeros, *Proc. 27th Ann. ACM Symp. on Theory of Computing*, 741–750, ACM Press, New York, May, 1995.

[P02] V. Y. Pan, Univariate Polynomials: Nearly Optimal Algorithms for Factorization and Rootfinding, *Journal of Symbolic Computations*, **33, 5**, 701-733, 2002.

[S98] G. W. Stewart, *Matrix Algorithms, Vol II: Eigensystems*, SIAM, Philadelphia, PA, 1998.

[VVMa] R. Vandebril, M. Van Barel, N. Mastronardi. An orthogonal similarity reduction of a matrix to semiseparable form. Technical Report TW355, *Katholieke Universiteit Leuven, Department Computerwetenschappen*, Leuven, Belgium, 2003.

Appendix. Figures

$$\begin{pmatrix} X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ 0 & 0 & 0 & X & X \\ X & X & X & X & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & X \\ X & X & X & X & X \\ X & X & X & X & X \\ 0 & 0 & 0 & X & X \\ X & X & X & X & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & X & X & X & X \\ X & X & X & X & X \\ 0 & 0 & X & 0 & X \\ 0 & 0 & 0 & X & X \\ X & X & X & X & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & X \\ X & X & X & X & X \\ 0 & 0 & X & 0 & X \\ 0 & 0 & 0 & X & X \\ X & X & X & X & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & X & X & X & X \\ 0 & X & 0 & 0 & X \\ 0 & 0 & X & 0 & X \\ 0 & 0 & 0 & X & X \\ X & X & X & X & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & X \\ 0 & X & 0 & 0 & X \\ 0 & 0 & X & 0 & X \\ 0 & 0 & 0 & X & X \\ X & X & X & X & X \end{pmatrix}$$

Figure 1:
Transforms: $T_1 \to G_{\mathbf{x_1}}(T_1) \to T_2 \to G_{\mathbf{x_2}}(T_2) \to T_3 \to G_{\mathbf{x_3}}(T_3) \to T_4 \to T_5 = G_{\mathbf{x_4}}(T_4)$.

$$\begin{pmatrix} X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \\ X & X & X & X & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & X & X & X & X \\ X & X & X & X & 0 \\ X & X & X & X & 0 \\ X & X & X & X & 0 \\ X & X & X & X & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & X & X & X & X \\ X & X & X & X & X \\ X & 0 & X & X & X \\ X & 0 & X & X & X \\ X & 0 & X & X & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & X & X & X & X \\ X & X & X & X & 0 \\ X & 0 & X & X & 0 \\ X & 0 & X & X & 0 \\ X & 0 & X & X & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & X & X & X & X \\ X & X & 0 & X & X \\ X & 0 & X & X & X \\ X & 0 & 0 & X & X \\ X & 0 & 0 & X & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & X & X & X & X \\ X & X & 0 & X & 0 \\ X & 0 & X & X & 0 \\ X & 0 & 0 & X & 0 \\ X & 0 & 0 & X & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & X & X & X & X \\ X & X & 0 & 0 & X \\ X & 0 & X & 0 & X \\ X & 0 & 0 & X & X \\ X & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & X & X & X & X \\ X & X & 0 & 0 & 0 \\ X & 0 & X & 0 & 0 \\ X & 0 & 0 & X & 0 \\ X & 0 & 0 & 0 & X \end{pmatrix}$$

Figure 2:
Transforms: $T^{(1)} \rightarrow G^{(\mathbf{y_1})}(T^{(1)}) \rightarrow T^{(2)} \rightarrow G^{(\mathbf{y_2})}(T^{(2)}) \rightarrow T^{(3)} \rightarrow G^{(\mathbf{y_3})}(T^{(3)}) \rightarrow T^{(4)} \rightarrow T^{(5)} = G^{(\mathbf{y_4})}(T^{(4)})$.

$$\begin{pmatrix} X & X & X & X & X \\ 0 & X & X & X & X \\ 0 & 0 & X & X & X \\ 0 & 0 & 0 & X & X \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & 0 \\ 0 & X & X & X & X \\ 0 & 0 & X & X & X \\ 0 & 0 & 0 & X & X \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & X & X & X & X \\ 0 & X & X & X & X \\ 0 & 0 & X & X & X \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & 0 \\ 0 & X & X & X & X \\ 0 & 0 & X & X & X \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & X & X & X & X \\ 0 & X & X & X & X \\ 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & 0 \\ 0 & X & X & X & X \\ 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & X & X & X & X \\ 0 & X & 0 & 0 & 0 \\ 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & 0 \\ 0 & X & 0 & 0 & 0 \\ 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix}$$

Figure 3:
Transforms: $R_1 \rightarrow G_{\mathbf{x_1}}(R_1) \rightarrow R_2 \rightarrow G_{\mathbf{x_2}}(R_2) \rightarrow R_3 \rightarrow G_{\mathbf{x_3}}(R_3) \rightarrow R_4 \rightarrow R_5 = G_{\mathbf{x_4}}(R_4)$.

$$\begin{pmatrix} X & X & X & X & X \\ 0 & X & X & X & X \\ 0 & 0 & X & X & X \\ 0 & 0 & 0 & X & X \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & X & X & X & 0 \\ 0 & X & X & X & 0 \\ 0 & 0 & X & X & 0 \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & 0 & X & X & X \\ 0 & X & X & X & X \\ 0 & 0 & X & X & X \\ 0 & 0 & 0 & X & X \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & X & X & 0 \\ 0 & X & X & X & 0 \\ 0 & 0 & X & X & 0 \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & 0 & 0 & X & X \\ 0 & X & 0 & X & X \\ 0 & 0 & X & X & X \\ 0 & 0 & 0 & X & X \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & X & 0 \\ 0 & X & 0 & X & 0 \\ 0 & 0 & X & X & 0 \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow$$

$$\Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & X \\ 0 & X & 0 & 0 & X \\ 0 & 0 & X & 0 & X \\ 0 & 0 & 0 & X & X \\ 0 & 0 & 0 & 0 & X \end{pmatrix} \Longrightarrow \begin{pmatrix} X & 0 & 0 & 0 & 0 \\ 0 & X & 0 & 0 & 0 \\ 0 & 0 & X & 0 & 0 \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & 0 & X \end{pmatrix}$$

Figure 4:
Transforms: $R^{(1)} \to G^{(\mathbf{y_1})}(R^{(1)}) \to R^{(2)} \to G^{(\mathbf{y_2})}(R^{(2)}) \to R^{(3)} \to G(R^{(3)}) \to R^{(4)} \to R^{(5)} = G^{(\mathbf{y_4})}(R^{(4)})$.