Advances in Linear Algebra & Matrix Theory, 2014, 4, 65-78
Published Online June 2014 in SciRes. http://www.scirp.org/journal/alamt
http://dx.doi.org/10.4236/alamt.2014.42004

New Method of Givens Rotations for Triangularization of Square Matrices

Artyom M. Grigoryan

Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, USA
Email: amgrigoryan@utsa.edu

Received 4 March 2014; revised 4 April 2014; accepted 14 April 2014

Abstract

This paper describes a new method of QR-decomposition of square nonsingular matrices (real or complex) by the Givens rotations through the unitary discrete heap transforms. This transforms can be defined by a different path, or the order of processing components of input data, which leads to different realizations of the QR-decomposition. The heap transforms are fast, because of a simple form of decomposition of their matrices. The direct calculation of the N-point discrete heap transform requires no more than 5(*N* − 1) multiplications, 2(*N* − 1) additions, plus 3(*N* − 1) trigonometric operations. The QR-decomposition of the square matrix *N* × *N* uses about 4/3*N*³ multiplications and *N*(*N* − 1)/2 square roots. This number varies depending on the path of the heap transform, and it is shown that "the optimal path" allows for significant reduction of number of operations in QR-decomposition. The heap transform and its matrix can be described analytically, and therefore, this transform can also be applied to the QR-decomposition of complex matrices.

Keywords

QR-Factorization, Givens Rotations, Householder Reflections, Heap Transform

1. Introduction

In linear algebra, methods of QR-decomposition (or factorization) of a nonsingular matrix into a unitary matrix and a triangular matrix are well known in mathematics [1]-[3]. QR-decomposition is used in many applications in computing and data analysis. This is the problem of solution of a linear system of equations written in matrix form as $\boldsymbol{Ax}=\boldsymbol{y}$. The solution $\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{y}$ can be found after the factorization of the matrix $\boldsymbol{A}=\boldsymbol{QR}$, where $\boldsymbol{Q}$ is an orthogonal matrix $\left(\boldsymbol{Q}^{-1}=\boldsymbol{Q}^{\mathrm{T}}\right)$ and $\boldsymbol{R}$ is a right triangular matrix, in the case when the dimensions of the known vector $\boldsymbol{y}$ and unknown $\boldsymbol{x}$ are equal. This QR-decomposition is unique if the diagonal coeffi-

How to cite this paper: Grigoryan, A.M. (2014) New Method of Givens Rotations for Triangularization of Square Matrices. *Advances in Linear Algebra & Matrix Theory*, **4**, 65-78. http://dx.doi.org/10.4236/alamt.2014.42004

cients of the matrix $\boldsymbol{R}$ are positive. In this case, $\boldsymbol{x} = \boldsymbol{R}^{-1}\boldsymbol{Q}\boldsymbol{y}$, or $\boldsymbol{x} = \boldsymbol{R}^{-1}\boldsymbol{P}\boldsymbol{Q}\boldsymbol{y}$ in the rank-revealing QR algorithm, when the diagonal elements of the matrix $\boldsymbol{R}$ are permuted $\boldsymbol{P}$ in the non-increasing order. There are several methods for computing the QR-decomposition, such as the Gramm-Schmidt process and method of Cholesky factorization. We here mention two other methods: the Householder transformations (known also as Householder reflections) and the Givens rotations. In the second method, each rotation zeros one element in the subdiagonal of the matrix. Therefore, a sequence of $N(N-1)/2$ plane rotations is required for reduction of a square matrix $(N \times N)$ to triangular form. The Givens rotations require a large number of arithmetical operations, including multiplications and $N(N-1)/2$ square roots [4]. The method of Householder transforms is the most applied method for QR-decomposition, which reduces the number of square roots to at most $2(N-1)$ and uses about $4N^3/3$ multiplications [5]-[8].

In this paper, a new look on the application of Givens rotations to the QR-decomposition problem is described, which is similar to the method of Householder transformations. The concept of the discrete heap transform is applied, which has been introduced in digital signal processing to generate the signal-induced unitary transforms by Grigoryan [9]-[12]. Both cases of real and complex matrices are considered and examples of performing the QR-decomposition of matrices are given. We also illustrate the importance of the path of the heap transform in such decomposition. The traditional way of consequently performing the rotations of data in natural order 1,2,3,... is not the best way, or path, in calculating the QR-decomposition. We briefly describe other more effective paths in QR-decomposition by the heap transforms and give a comparison with the known method of the Householder transformation.

2. Transforms with Decision Equations

Let $f(x,y,\varphi)$ and $g(x,y,\varphi)$ be functions of three variables; φ is referred to as the rotation parameter such as the angle, and *x* and *y* as the coordinates of the point (x,y) on the plane. These variables may have other meanings. The function $g(x,y,\varphi)$ is parameterized and it is assumed that for a specified set of numbers a and for each point (x,y) on the plane (or its chosen subset) the equation

$$g(x,y,\varphi) = a \tag{1}$$

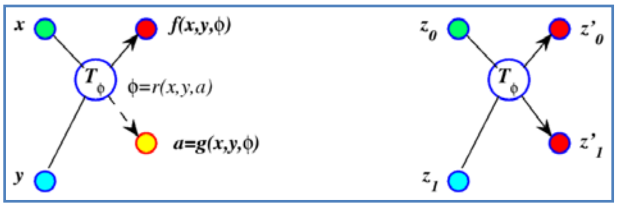
has a unique solution with respect to φ. We denote the solution of this equation by $\varphi = r(x,y,a)$. The system of equations $f(x,y,\varphi) = y_0, g(x,y,\varphi) = a,$ is called the *system of decision equations*. The value of φ is calculated from the second equation which is called the *angular equation*. Then the value of y_0 is calculated from the given input (x,y) and φ. It is also assumed that the two-dimensional transformation

$$T_\varphi : (z_0, z_1) \to (z_0', z_1') = \left(f(z_0, z_1, \varphi), g(z_0, z_1, \varphi)\right)$$

which is derived (or whose matrix is calculated) from the given decision equations by $T_\varphi : (z_0, z_1) \to \left(f(z_0, z_1, \varphi), a\right)$ is unitary. These transformations are basic stones to build unitary transforms.

Figure 1 shows the graph of such a basic transformation in part a. The transformation T_φ is defined by given x and y, namely, the angle φ is calculated first and then the transform $T_\varphi(x,y)$ is calculated. The second output is shown with the dash arrow, since the value of this output equals a which is known from given angular Equation (1). The graph of this transform applied to an input (z_0, z_1) is shown in part b.

Example 1: (*Elementary rotation*). Given a real number a, we consider the following functions:



(a) Y-operator (b) X-operator

Figure 1. Graphs of the basic transform T_φ defined (a) by a point (x,y) and applied (b) on an input (z_0, z_1).

$$f(x,y,\varphi)=x\cos\varphi-y\sin\varphi,\quad g(x,y,\varphi)=x\sin\varphi+y\cos\varphi.$$

The basic transformation is defined as a rotation of the point (x,y) to the horizontal $Y=a$,

$$T_\varphi:(x,y)\to(x\cos\varphi-y\sin\varphi,a),$$

where the rotation angle φ is calculated by $\varphi=\cos^{-1}\left(a/\sqrt{x^2+y^2}\right)-\tan^{-1}x/y$ and $\varphi=\sin^{-1}(a/x)$ if $y=0$. The angular equation puts a constrain on parameter a, namely, it is required that $a^2\le x^2+y^2$.

3. Discrete Heap Transforms

The composition of the *N*-dimensional discrete heap transform by a given vector-generator $\boldsymbol{x}=(x_0,x_1,x_2,\cdots,x_{N-1})'$ is performed by the sequential calculation of basic two-dimensional transformations. In matrix form, this composition can be written as follows: $\boldsymbol{H}=\boldsymbol{T}_1\boldsymbol{T}_2\boldsymbol{T}_3\cdots\boldsymbol{T}_{N-1}$ where each transformation T_k, $k=1:(N-1)$, changes only two components of the input. We assume that the components of an input $\boldsymbol{z}=(z_0,z_1,z_2,\cdots,z_{N-1})'$ are processed in order $z_0,z_1,\cdots,$ and z_{N-1}. This is a natural path P, and in general, such a path can be taken in many different ways. It is a very important characteristic of the heap transform and the right selection of the path leads to an effective application of the transform in calculating the QR-decomposition. We consider the case, when all basic transformations T_k are parameterized by angles φ_k. The special selection of a set of parameters is initiated by the vector-generator $\boldsymbol{x}$ through the decision equations with a given set of constants $A=\{a_1,a_2,\cdots,a_{N-1}\}$. The generator is processed first, and during this process all required angles are calculated. For a given set A, we define the following "heap" of elements:

$$y_0^{(1)}=f(x_0,x_1,\varphi_1),y_0^{(2)}=f\left(y_0^{(1)},x_2,\varphi_2\right),y_0^{(3)}=f\left(y_0^{(2)},x_3,\varphi_3\right),\cdots,y_0^{(N-1)}=f\left(y_0^{(N-2)},x_{N-1},\varphi_{N-1}\right),$$

where angles φ_k are calculated by $\varphi_k=r\left(y_0^{(k)},x_k,a_k\right)$, $\left(y_0^{(0)}=x_0\right)$, and $k=1:(N-1)$. Thus, it is assumed that the first component of the vector participates in calculations of all $(N-1)$ basic transforms T_k. As an example, **Figure 2** shows the signal-flow graph of determination of the five-point transform by a vector $\boldsymbol{x}$.

The *N*-point transformation T composing from T_k in the space of *N*-dimensional vectors $\boldsymbol{z}$ is defined as

$$\boldsymbol{z}=(z_0,z_1,z_2,\cdots,z_{N-1})'\to T(\boldsymbol{z})=\left(z_0^{(N-1)},z_1^{(1)},z_2^{(1)},\cdots,z_{N-1}^{(1)}\right).$$

Values of the components $z_k^{(1)}$ are calculated by

$$z_k^{(1)}=f\left(z_0^{(k-1)},z_k,\varphi_k\right),\ k=1:(N-1),$$

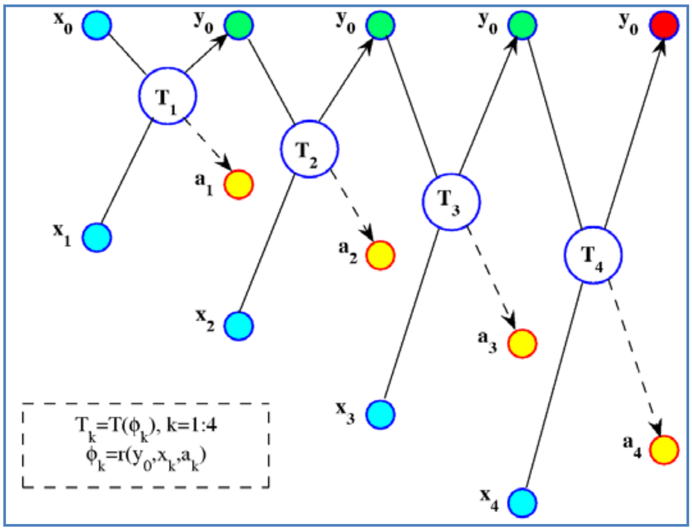


Figure 2. Signal-flow graph of determination of the five-point transform by a vector $x=(x_0,x_1,x_2,x_2,x_4)'$.

where $z_0^{(0)} = z_0$. The first component is transformed sequentially as $z_0 \to z_0^{(1)} \to z_0^{(2)} \to \cdots \to z_0^{(N-1)}$ where $z_0^{(1)} = f\left(z_0, z_1, \varphi_1\right)$, $z_0^{(2)} = f\left(z_0^{(1)}, z_2, \varphi_2\right)$, $\cdots$, $z_0^{(N-1)} = f\left(z_0^{(N-2)}, z_{N-1}, \varphi_{N-1}\right)$. The following should be mentioned about the heap transform. The transform is performed in a space of *N*-dimensional vectors $\boldsymbol{z}$, but all angles φ_k are found and the transformation T is composed after solving the decision equations relative to a given vector-generator $\boldsymbol{x}$. The transform of $\boldsymbol{x}$ results in a vector with the constant components of the set *A*, plus $y_0^{(N-1)}$ as the first component,

$$\boldsymbol{x} \to T\left(\boldsymbol{x}\right) = \left(y_0^{(N-1)}, a_1, a_2, \cdots, a_{N-1}\right)'.$$

The transformation T is called the *N*-point discrete $\boldsymbol{x}$*-signal-induced heap transformation* (DsiHT), and the vector $\boldsymbol{x}$ is *the generator* of this transformation [6]. The first component $y_0^{(N-1)}$ is referred to as *the heap* of the transform.

3.1. Transform Collecting Energy

We focus on the special case of the DsiHT, namely on transformations that collect the energy of vectors in one location. The angular equations for such transformations are defined by the set $A = \{0\}$, *i.e.*, when $a_k = 0$, $k = 1:(N-1)$. This condition leads to the fact that the first basis functions of such transforms are the vector-generators themselves. Other basis functions are defined based on the correlation data of components of these vectors. Matrices of these transforms are orthogonal. The transforms have simple forms of decomposition that lead to calculation of the *N*-point transforms with no more than $7(N-1)$ multiplications and $3(N-1)$ additions. $3(N-1)$ operations are also required for calculating elementary trigonometric functions. In the linear space of *N*-dimension vectors, we construct such an orthogonal transform, T, whose matrix is defined by $\boldsymbol{Tx} = E\boldsymbol{e}$ where E denotes the energy of the vector and $\boldsymbol{e}$ is the unit vector $(1, 0, \cdots, 0)'$. The term "energy" is referred to as the norm of the vector, *i.e.*, $E^2 = \|\boldsymbol{x}\|^2 = x_0^2 + x_1^2 + \cdots + x_{N-1}^2$. To define the desired transform T, we consider the following method of energy transferring. Let $\bar{x} = \left(x_0, x_1\right)'$ be a 2-D vector to be rotated into a vector $\bar{y} = \left(y_0, 0\right)'$, *i.e.*, $T(\varphi)\bar{x} = \bar{y}$.

The matrix of the transformation, $T(\varphi)$, which is the Givens rotation, is defined by the matrix equation

$$\bar{y} = \begin{bmatrix} y_0 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}. \tag{2}$$

The angle φ and component y_0 are calculated as $\varphi = -\tan^{-1}\left(x_1/x_0\right)$ if $x_0 \neq 0$, and $y_0 = \sqrt{x_0^2 + x_1^2} = x_0/\cos\varphi$. If $x_0 = 0$, then $\varphi = \pi/2$ and $y_0 = -x$. The energy is collected in the first components, $y_0^2 = x_0^2 + x_1^2$.

When processing the *N*-dimensional vector $\boldsymbol{x}$, the first pair of components, $\bar{x}_0 = \left(x_0, x_1\right)'$, is transferred to the vector $\bar{y} = \left(y_0, 0\right)'$ by Equation (2), as the point $\left(x_0, x_1\right)$ is rotated to the point $\left(y_0, 0\right)$ on the horizontal line. Then, on the next *k*th step of calculations, when $k \geq 1$, the similar rotations are accomplished over vectors $\bar{x}_{k-1} = \left(y_0^{(k-1)}, x_k\right)'$, where $y_0^{(k-1)}$ denotes a new value of the first component on the (*k* − 1)th step, and $y_0^{(0)} = x_0$. The value of the first component y_0 is renewed consequently as

$$\begin{bmatrix} y_0^{(k)} \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\varphi_k & -\sin\varphi_k \\ \sin\varphi_k & \cos\varphi_k \end{bmatrix} \begin{bmatrix} y_0^{(k-1)} \\ x_k \end{bmatrix} \tag{3}$$

where the angles are defined by

$$\tan\varphi_k = -x_k / y_0^{(k-1)},\ y_0^{(k-1)} = \left(\boldsymbol{T}_{k-1}\cdots\boldsymbol{T}_2\boldsymbol{T}_1 x\right)_0 = \sqrt{x_0^2 + x_1^2 + \cdots + x_{k-1}^2},\ \ k = 1:(N-1). \tag{4}$$

Here, $\boldsymbol{T}_m$ are respectively the matrices of the Givens rotations $T_m = T\left(\varphi_m\right)$, $m = 1:(k-1)$. As a result, the whole energy of the input signal is collected consequently and, then, transferred to the first component

$$y_0 = y_0^{(N-1)} = \sqrt{x_0^2 + x_1^2 + \cdots + x_{N-1}^2}$$

of the transform $\boldsymbol{y} = \left(y_0, 0, \cdots, 0\right)'$. **Figure 3** shows the signal-flow graph of calculation of the four-point trans-

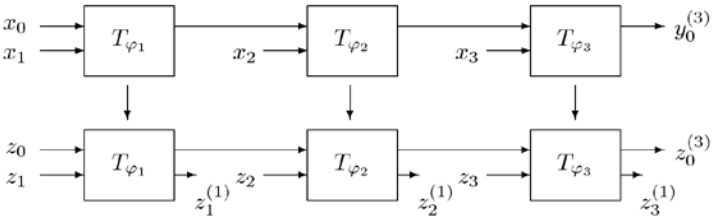


Figure 3. Signal-flow graph of calculation of the four-point heap transform of the vector $\boldsymbol{z}$.

form of a vector $\boldsymbol{z}=(z_0,z_1,z_2,z_3)'$. It is assumed that all parameters φ_k, $k=1,2,3$, are calculated by the given vector $\boldsymbol{x}=(x_0,x_1,x_2,x_3)'$ and then used when transforming the vector $\boldsymbol{z}\to T[\boldsymbol{z}]$.

3.2. Analytical Expressions

It is important to note, that the basic transformations, $T_k=T(\varphi_k)$, $k=1:(N-1)$, that compose the N-point DsiHT, can be performed without calculation of angles and trigonometric functions. Analytical formulas can be derived for calculation of components of the heap transform $T:(z_0,z_1,z_2,\cdots,z_{N-1})\to\left(z_0^{(N-1)},z_1^{(1)},z_2^{(1)},\cdots,z_{N-1}^{(1)}\right)$. The matrix of the transform can also be calculated analytically. To show this fact, we introduce the following notations which represent respectively the partial cross-correlation of $\boldsymbol{z}$ with the vector-generator $\boldsymbol{x}$ and energy of $\boldsymbol{x}$:

$$E_k(\boldsymbol{z},\boldsymbol{x})=z_0x_0+z_1x_1+\cdots+z_{k-1}x_{k-1},\quad E_k^2(\boldsymbol{x})=E_k(\boldsymbol{x},\boldsymbol{x})=x_0^2+x_1^2+\cdots+x_{k-1}^2,$$

where $k=1:(N-1)$. The components of the heap transform on the kth iteration can be expressed by correlation data as

$$z_0^{(k)}=\frac{E_{k+1}(\boldsymbol{z},\boldsymbol{x})}{E_{k+1}(\boldsymbol{x})},\quad z_k^{(1)}=-\frac{E_k(\boldsymbol{z},\boldsymbol{x})x_k-z_kE_k^2(\boldsymbol{x})}{E_{k+1}(\boldsymbol{x})E_k(\boldsymbol{x})},\quad k=1:(N-1). \tag{5}$$

On the final step, the value of the first component is defined by $z_0^{(N-1)}=E_N(\boldsymbol{z},\boldsymbol{x})/E_N(\boldsymbol{x})$ which is the correlation coefficient of the input signal $\boldsymbol{z}$ with the normalized signal-generator $\boldsymbol{x}$. In the particular $\boldsymbol{z}=\boldsymbol{x}$ case, we obtain $z_0^{(N-1)}=E_N(\boldsymbol{x})=\|\boldsymbol{x}\|$ and the rest of coefficients $z_k^{(1)}=0, k=1:(N-1)$.

The coefficients $t_{n,m}$ of the matrix of the N-point DsiHT can be obtained from equations in (5). The mth row of the matrix of the transform is defined by applying the unit vector $\boldsymbol{z}=\boldsymbol{e}_m=(0,0,\cdots,1,\cdots,0)'$ with 1 on the mth position, where integer $m\in[0,N-1]$. Therefore, the coefficients can be calculated as

$$t_{0,m}=\frac{E_N(\boldsymbol{e}_m,\boldsymbol{x})}{E_N(\boldsymbol{x})},\quad t_{n,m}=-\frac{E_n(\boldsymbol{e}_m,x)x_n-(\boldsymbol{e}_m)_nE_n^2(\boldsymbol{x})}{E_n(\boldsymbol{x})E_{n+1}(\boldsymbol{x})},\quad m=0:(N-1),n=1:(N-1). \tag{6}$$

4. Transforms with Decision Equations

In this section, we give a brief analysis of the matrix decomposition by the heap transformations and the comparison with the known methods which are based on rotations in two dimensions. It should be noted first that the discrete heap transformations differ from the well-known Householder transformation [2], whose matrix $(N\times N)$ is symmetric and defined as $\boldsymbol{H}=\boldsymbol{I}-2\boldsymbol{w}\boldsymbol{w}'$, where the normalized vector $\boldsymbol{w}$ is calculated

$$\boldsymbol{w}=\boldsymbol{x}-(\|\boldsymbol{x}\|,0,0,\cdots,0)'\Big/\sqrt{2\|\boldsymbol{x}\|(\|\boldsymbol{x}\|-x_0)}$$

and $\boldsymbol{x}=(x_0,x_1,x_2,\cdots,x_{N-1})'$ is a given vector. For example, when $\boldsymbol{x}=(1,2,3,4)'$, the Householder vector equals $\boldsymbol{w}=(-0.6393,0.2856,0.4284,0.5712)'$ and the matrix of the Householder transform is

$$\boldsymbol{H}=\begin{bmatrix}0.1826 & 0.3651 & 0.5477 & 0.7303\\ 0.3651 & 0.8369 & -0.2447 & -0.3262\\ 0.5477 & -0.2447 & 0.6330 & -0.4893\\ 0.7303 & -0.3262 & -0.4893 & 0.3475\end{bmatrix}.$$

The matrix of the heap transformation generated by the same vector x equals

$$\boldsymbol{T}=\begin{bmatrix} 0.1826 & 0.3651 & 0.5477 & 0.7303 \\ -0.8944 & 0.4472 & 0 & 0 \\ -0.3586 & -0.7171 & 0.5976 & 0 \\ -0.1952 & -0.3904 & -0.5855 & 0.6831 \end{bmatrix}.$$

These two transformations result in the same vector, $\boldsymbol{Hx}=\boldsymbol{Tx}=(5.4772,0,0,0)'$, where $5.4772=\|\boldsymbol{x}\|$. Both transformations have the same first basis function which equals the normalized vector-generator.

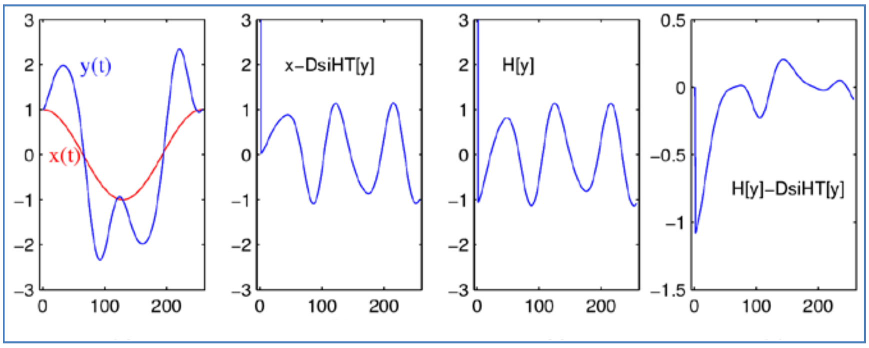
The heap and Householder transformations differ much when considered with the large dimensions. For instance, when $N=7$ and the vector-generator $\boldsymbol{x}=(1,2,3,4,-3,-2,-1)'$, we obtain the following symmetric matrix of the Householder transformation with determinant one:

$$\boldsymbol{H}=\begin{bmatrix} 0.1508 & 0.3015 & 0.4523 & 0.6030 & -0.4523 & -0.3015 & -0.1508 \\ 0.3015 & 0.8930 & -0.1606 & -0.2141 & 0.1606 & 0.1070 & 0.0535 \\ 0.4523 & -0.1606 & 0.7591 & -0.3211 & 0.2409 & 0.1606 & 0.0803 \\ 0.6030 & -0.2141 & -0.3211 & 0.5718 & 0.3211 & 0.2141 & 0.1070 \\ -0.4523 & 0.1606 & 0.2409 & 0.3211 & 0.7591 & -0.1606 & -0.0803 \\ -0.3015 & 0.1070 & 0.1606 & 0.2141 & -0.1606 & 0.8930 & -0.0535 \\ -0.1508 & 0.0535 & 0.0803 & 0.1070 & -0.0803 & -0.0535 & 0.9732 \end{bmatrix}$$

and the vector $\boldsymbol{w}=(-0.6516,0.2314,0.3470,0.4627,-0.3470,-0.2314,-0.1157)'$. The orthogonal matrix of the heap transformation generated by the vector $\boldsymbol{x}$ is not full filled and has the zero upper triangular submatrix with 15 zeros,

$$\boldsymbol{T}=\begin{bmatrix} 0.1508 & 0.3015 & 0.4523 & 0.6030 & -0.4523 & -0.3015 & -0.1508 \\ -0.8944 & 0.4472 & 0 & 0 & 0 & 0 & 0 \\ -0.3586 & -0.7171 & 0.5976 & 0 & 0 & 0 & 0 \\ -0.1952 & -0.3904 & -0.5855 & 0.6831 & 0 & 0 & 0 \\ 0.0877 & 0.1754 & 0.2631 & 0.3508 & 0.8771 & 0 & 0 \\ 0.0488 & 0.0977 & 0.1465 & 0.1954 & -0.1465 & 0.9524 & 0 \\ 0.0230 & 0.0460 & 0.0690 & 0.0920 & -0.0690 & -0.0460 & 0.9886 \end{bmatrix}.$$

We now illustrate the difference of the DsiHT and Householder transforms applied on signals. **Figure 4** shows the example with the 257-point signal $x(t)=\cos(2t)\pi, t\in[0,\ ]$ and noisy signal $y(t)=2x(t)-\cos(6t)+0.2\sin(10t)$ in part a, along with the $x(t)$-induced heap and Householder transforms



(a) (b) (c) (d)

Figure 4. (a) The original $x(t)$ and signal $y(t)$ with a random noise, (b) the 257-point discrete heap transform of $y(t)$, (c) the 257-point Householder transform of $y(t)$, and (d) the point wise difference of these transforms.

of $y(t)$ respectively in b and c, and the difference of these two transforms in *d*. The first seven values of the Householder and heap transforms are the following:

$$\boldsymbol{Hy}: \quad 22.6276, \quad -1.0608, \quad -1.0285, \quad -0.9915, \quad -0.9503, \quad -0.9052, \quad -0.8568,$$
$$\boldsymbol{Ty}: \quad 22.6276, \quad 0.0190, \quad 0.0366, \quad 0.0566, \quad 0.0788, \quad 0.1031, \quad 0.1291.$$

The big difference of these transforms occurs in the first ninety components.

5. Matrix Decomposition

We now consider the application of heap transforms for calculating the determinant of a nonsingular square matrix, as well as its decomposition. The method described in this section is based on the same idea as the method of Householder transforms, also known as the Householder reflections. Although the heap transformation and Householder transformation are different, it is evident that these two concepts should result in the same (or similar) decomposition, but by different ways.

Let $\boldsymbol{X}$ be a real square matrix ($N \times N$) with $\det(\boldsymbol{X}) \neq 0$. We denote coefficients of this matrix as

$$\boldsymbol{X} = \begin{bmatrix} a_0 & b_0 & c_0 & \cdots & d_0 \\ a_1 & b_1 & c_1 & \cdots & d_1 \\ a_2 & b_2 & c_2 & \cdots & d_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{N-1} & b_{N-1} & c_{N-1} & \cdots & d_{N-1} \end{bmatrix}.$$

Let vector $\boldsymbol{a}$ be the first column of this matrix, $\boldsymbol{a} = (a_0, a_1, a_2, \cdots, a_{N-1})'$. We denote by $\boldsymbol{T_a}$ the matrix of the heap transformation $T_{\boldsymbol{a}}$ generated by this vector. This matrix contains an upper triangular submatrix with $N(N-3)/2+1$ zeros. The product of two matrices $\boldsymbol{X}_1 = \boldsymbol{T_a X}$ results in a matrix $(N \times N)$ with the first column equal to $\bar{\boldsymbol{a}} = T_{\boldsymbol{a}}(\boldsymbol{a}) = (\|\boldsymbol{a}\|, 0, 0, \cdots, 0)'$. Therefore, the matrix $\boldsymbol{X}_1$ has the following form:

$$\boldsymbol{X}_1 = \begin{bmatrix} \|\boldsymbol{a}\| & \bar{b}_0 & \bar{c}_0 & \cdots & \bar{d}_0 \\ 0 & \bar{b}_1 & \bar{c}_1 & \cdots & \bar{d}_1 \\ 0 & \bar{b}_2 & \bar{c}_2 & \cdots & \bar{d}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \bar{b}_{N-1} & \bar{c}_{N-1} & \cdots & \bar{d}_{N-1} \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{a}\| & \bar{b}_0 & \bar{c}_0 & \cdots & \bar{d}_0 \\ 0 & & & & \\ 0 & & \boldsymbol{Y}_1 & & \\ \vdots & & & & \\ 0 & & & & \end{bmatrix}$$

where we denote by $\boldsymbol{Y}_1$ the $(1,1)$ minor of this product. We can repeat the described above process for the submatrix $\boldsymbol{Y}_1$, by considering its first vector-column $\boldsymbol{b} = (\bar{b}_0, \bar{b}_1, \bar{b}_2, \cdots, \bar{b}_{N-1})'$ as a generator for the $(N-1)$-point heap tranformation $T_{\boldsymbol{b}}$, whose matrix we denote by $\boldsymbol{T_b}$. The product of two matrices

$$\boldsymbol{X}_2 = \begin{bmatrix} 1 & 0 \\ 0 & \boldsymbol{T_b} \end{bmatrix} \boldsymbol{X}_1 = (1 \oplus \boldsymbol{T_b}) \boldsymbol{T_a X}$$

is a matrix ($N \times N$) with the second column equal

$$\bar{\boldsymbol{b}} = \begin{pmatrix} \bar{b}_0 \\ \boldsymbol{T_b}(\boldsymbol{b}) \end{pmatrix} = (\bar{b}_0, \|\boldsymbol{b}\|, 0, 0, \cdots, 0)'.$$

On this step, the matrix $\boldsymbol{X}$ is transformed to the matrix $\boldsymbol{X}_2$ which has the following form:

$$\boldsymbol{X}_2 = \begin{bmatrix} \|\boldsymbol{a}\| & \bar{b}_0 & \bar{c}_0 & \cdots & \bar{d}_0 \\ 0 & \|\boldsymbol{b}\| & \cdots & \cdots & \cdots \\ 0 & 0 & & & \\ \cdots & \cdots & & \boldsymbol{Y}_2 & \\ 0 & 0 & & & \end{bmatrix}$$

where $\boldsymbol{Y}_2$ is the (2,2) minor of this product and has the size $(N-2) \times (N-2)$.

On the next step, we consider the first vector c of the submatrix $\boldsymbol{Y}_2$ and construct the matrix $\boldsymbol{T_c}$ of heap transformation generated by $\boldsymbol{c}$. Then, the (3,3) minor $\boldsymbol{Y}_3$ of the product of the matrices

$$\boldsymbol{X}_3=\left(1\oplus1\oplus\boldsymbol{T_c}\right)\boldsymbol{X}_2=\left(1\oplus1\oplus\boldsymbol{T_c}\right)\left(1\oplus\boldsymbol{T_b}\right)\boldsymbol{T_a}\boldsymbol{X}$$

is transformed into a matrix whose the first column is zero, except in the first row. The process of such transformations is repeated until we obtain the upper triangular matrix:

$$\boldsymbol{X}_{N-1}=\left(1\oplus\cdots\oplus1\oplus\boldsymbol{T_d}\right)\cdots\left(1\oplus1\oplus\boldsymbol{T_c}\right)\left(1\oplus\boldsymbol{T_b}\right)\boldsymbol{T_a}\boldsymbol{X}=\begin{bmatrix}\|\boldsymbol{a}\| & \bar{b}_0 & \bar{c}_0 & \cdots & \bar{d}_0\\ 0 & \|\boldsymbol{b}\| & \cdots & \cdots & \cdots\\ 0 & 0 & \|\boldsymbol{c}\| & \cdots & \cdots\\ \cdots & \cdots & \cdots & \cdots & \cdots\\ 0 & 0 & 0 & \cdots & \|\boldsymbol{d}\|\end{bmatrix}.$$

The determinant of the matrix equals $\det\boldsymbol{X}_{N-1}=\det\boldsymbol{X}=\|\boldsymbol{a}\|\cdot\|\boldsymbol{b}\|\cdot\|\boldsymbol{c}\|\cdots\|\boldsymbol{d}\|$, since all heap transformations have determinant one. Thus, the determinant of a square nonsingular matrix can be calculated by using the heap transforms. We denote by $\boldsymbol{R}$ the product of the following matrices

$$\boldsymbol{R}=\left(1\oplus\cdots\oplus1\oplus\boldsymbol{T_d}\right)\cdots\left(1\oplus1\oplus\boldsymbol{T_c}\right)\left(1\oplus\boldsymbol{T_b}\right)\boldsymbol{T_a}.$$

Each heap transformation is unitary, and R is unitary. The inverse matrix $\boldsymbol{R}^{-1}$ can thus be represented by

$$\boldsymbol{R}^{-1}=\boldsymbol{R}'=\boldsymbol{T}'_{\boldsymbol{a}}\left(1\oplus\boldsymbol{T}'_{\boldsymbol{b}}\right)\left(1\oplus1\oplus\boldsymbol{T}'_{\boldsymbol{c}}\right)\cdots\left(1\oplus\cdots\oplus1\oplus\boldsymbol{T}'_{\boldsymbol{d}}\right).$$

We obtain the following decompositions of the matrix $\boldsymbol{X}$ and its inverse: $\boldsymbol{X}=\boldsymbol{R}^{-1}\boldsymbol{X}_{N-1}$, $\boldsymbol{X}^{-1}=\boldsymbol{X}_{N-1}^{-1}\boldsymbol{R}$, where $\boldsymbol{X}_{N-1}$ and $\boldsymbol{X}_{N-1}^{-1}$ are the upper triangular matrices, and $\boldsymbol{R}$ is unitary.

Example 2: Consider the decomposition of the following matrix (3 × 3):

$$\boldsymbol{X}=\begin{bmatrix}12 & -51 & 4\\ 6 & 167 & -68\\ -4 & 24 & -41\end{bmatrix},\qquad \det\left(\boldsymbol{X}\right)=-85750.$$

The decomposition of $\boldsymbol{X}$ by the heap transforms, $\boldsymbol{X}=\boldsymbol{R}'\boldsymbol{X}_2$, results in the unitary matrix

$$\boldsymbol{R}'=\begin{bmatrix}6/7 & -69/175 & 58/175\\ 3/7 & 158/175 & -6/175\\ -2/7 & 6/35 & 33/35\end{bmatrix}=\begin{bmatrix}0.85714286 & -0.39428571 & 0.33142857\\ 0.42857143 & 0.90285714 & -0.03428571\\ -0.28571429 & 0.17142857 & 0.94285714\end{bmatrix}$$

and the upper triangular matrix

$$\boldsymbol{X}_2=\begin{bmatrix}14 & 21 & -14\\ 0 & 175 & -70\\ 0 & 0 & -35\end{bmatrix}.$$

The coefficients of the matrix $\boldsymbol{R}'$ are given with precision of eight digits after the point. The decomposition of the matrix $\boldsymbol{X}=\boldsymbol{Q}_H\boldsymbol{R}_H$ by the Householder transforms is defined by the same matrices, $\boldsymbol{Q}_H=\boldsymbol{R}'$ and $\boldsymbol{R}_H=\boldsymbol{X}_2$. Thus, by means of the heap transforms which are composed by the Givens rotations, we achieve the same decomposition as the Householder reflections. It should be noted, that the first row-vector of the matrix $\boldsymbol{R}$ equals to the normalized first column-vector $\boldsymbol{a}$ of the original matrix $\boldsymbol{X}$. For the example considered above, we obtain

$$\left(0.85714286,0.42857143,-0.28571429\right)'=\left(12,6,-4\right)'\Big/\sqrt{196}=\boldsymbol{a}/\|\boldsymbol{a}\|.$$

Arithmetic Complexity of the Decomposition

The $\boldsymbol{x}$-generated heap transform of a vector $\boldsymbol{z}$ can be calculated by using analytical Equations (5) and (6),

$$z_0^{(N-1)}=\frac{E_N\left(\boldsymbol{z},\boldsymbol{x}\right)}{E_N\left(\boldsymbol{x}\right)},\quad z_k^{(1)}=-\frac{E_k\left(\boldsymbol{z},\boldsymbol{x}\right)x_k-z_kE_k^2\left(\boldsymbol{x}\right)}{E_{k+1}\left(\boldsymbol{x}\right)E_k\left(\boldsymbol{x}\right)},\quad k=1:\left(N-1\right).\tag{7}$$

The calculation of all energy coefficients $E_n(\boldsymbol{x})$, $n=0:(N-1)$, requires N multiplications. The total number of multiplications is calculated by $\mu_{z;a}(N)=(N+(N-1))+(N+1+3(N-1))=6N-3$, plus N square roots. When the matrix $\boldsymbol{T}$ is multiplying by another square matrix, the number of required multiplications can be calculated as $\mu_{p;a}(N)=(N+(N-1))+(N-1)(4N-2)=4N^2-4N+1$ plus N square roots. If we use this estimation to calculate the number of multiplications for factorization of a square nonsingular matrix by heap transforms, we obtain the following number:

$$\mu_{ru;a}=\sum_{n=1}^{N}\mu_{p;a}(n)=\sum_{n=1}^{N}\left[4n^2-4n+1\right]=4N^3/3+N/2-2$$

plus $N(N-1)/2$ square roots.

6. Strong Heap Transform

The point, or the heap, where the energy of the vector-generator is transferred is not required to be at the first location of the signal. The DsiHT can be defined by equation $T\boldsymbol{x}=E\boldsymbol{e}$ for any unit vector $\boldsymbol{e}=(0,0,\cdots,1,\cdots,0)'$. The path *P* along which we compose the heap is important, not a location of energy integration, when defining the discrete heap transform. As an example, we consider the basic transforms that sequentially process the last pair of components of the signal. This is the case, when the path *P* of the heap transformation induced by a vector-signal *x* is defined as $P(\boldsymbol{x})_k=x_{N-1-k}$, $k=0:(N-1)$. We call this heap transformation *the strong heap transformation*. The energy from each component x_k is taken away and given to the last component x_{N-1}. The strong heap transformation generated by the vector x results in the transform $T_{\boldsymbol{x}}[\boldsymbol{x}]=(0,0,\cdots,0,\|x\|)'$.

Example 3: For the vector $\boldsymbol{x}=[1,2,3,4]'$, the matrix of this heap transformation is orthogonal and equals

$$\boldsymbol{T}=\begin{bmatrix}0.1826 & 0.3651 & 0.5477 & 0.7303\\ -0.9832 & 0.0678 & 0.1017 & 0.1356\\ 0 & -0.9285 & 0.2228 & 0.2971\\ 0 & 0 & -0.8000 & 0.6000\end{bmatrix}.$$

It is interesting to note that the energy of the signal is transferred sequentially to the first component in numbers 5, 5.3852, and 5.4772. This heap of numbers is smaller than the numbers 2.2361, 3.7417, and 5.4772, when the four-point DsiHT is composed along the ordinary path $P(\boldsymbol{x})_k=x_k$, $k=1,2,3$, by Equations (3) and (4). That can be explained because of small values of the first components of the signal $\boldsymbol{x}$.

The method of triangularization of a square nonsingular matrix $\boldsymbol{X}$ by the strong heap transformations is described similarly to the method described in Section 4 for the ordinary path. The only difference is that on each *k*th step of calculations, where $k=1:(N-1)$, the generator of the strong heap transformation, which is zeroing the *k*th column of the renewed matrix $\boldsymbol{X}_k$, is taken from the top part of the column. To illustrate this method, we consider the matrix

$$\boldsymbol{X}=\begin{bmatrix}1 & 2 & 3\\ 2 & 3 & 2\\ 3 & 4 & 4\end{bmatrix}.$$

The first column $\boldsymbol{a}=(1,2,3)'$ generates the strong heap transformation whose matrix equals

$$\boldsymbol{T}_{\boldsymbol{a}}=\begin{bmatrix}0.9636 & -0.1482 & -0.2224\\ 0 & 0.8321 & -0.5547\\ 0.2673 & 0.5345 & 0.8018\end{bmatrix}$$

and $\boldsymbol{T}_{\boldsymbol{a}}(\boldsymbol{a})=(0,0,3.7417)'$. The product of matrices $\boldsymbol{T}_{\boldsymbol{a}}$ and $\boldsymbol{X}$ equals

$$\boldsymbol{X}_1=\boldsymbol{T}_{\boldsymbol{a}}\boldsymbol{X}=\begin{bmatrix}0 & 0.5930 & 1.7049\\ 0 & 0.2774 & -0.5547\\ 3.7417 & 5.3452 & 5.0780\end{bmatrix}.$$

Continuing zeroing the first two elements of the second column of this matrix, by using the strong heap transformation defined by the vector $(0.5930,0.2774)'$, we obtain the following decomposition of the matrix

$\boldsymbol{X}$ by the product of the unitary matrix $\boldsymbol{R}'$ and the lower triangular matrix $\boldsymbol{X}_2$:

$$\boldsymbol{X} = \boldsymbol{R}^{-1}\boldsymbol{X}_2 = \begin{bmatrix} 0.4082 & 0.8729 & 0.2673 \\ -0.8165 & 0.2182 & 0.5345 \\ 0.4082 & -0.4364 & 0.8018 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1.2247 \\ 0 & 0.6547 & 1.3093 \\ 3.7417 & 5.3452 & 5.0780 \end{bmatrix}. \qquad (8)$$

The determinant of $\boldsymbol{X}$ can be calculated as $\det(\boldsymbol{X}) = -(3.7417 \cdot 0.6547 \cdot 1.2247) = -3$. Thus, we obtain the decomposition of the square matrix $\boldsymbol{X}$ by the product of the unitary matrix $\boldsymbol{R}'$ and lower triangular matrix $\boldsymbol{X}_2$. It should be noted for the comparison that when using the ordinary path, $P(\boldsymbol{x})_k = x_k$, $k = 0:(N-1)$, the following decompositions of the matrix $\boldsymbol{X}$ holds:

$$\boldsymbol{X} = \boldsymbol{R}^{-1}\boldsymbol{X}_2 = \begin{bmatrix} 0.2673 & -0.8729 & 0.4082 \\ 0.5345 & -0.2182 & -0.8165 \\ 0.8018 & 0.4364 & 0.4082 \end{bmatrix} \begin{bmatrix} 3.7417 & 5.3452 & 5.0780 \\ 0 & -0.6547 & -1.3093 \\ 0 & & 1.2247 \end{bmatrix}.$$

It is evident that there is a relation between this decomposition and the decomposition of Equation (8). The first and third columns of the matrices $\boldsymbol{R}'$ have been rearranged, as well as the first and third rows of the matrices $\boldsymbol{X}_2$. In addition, the signs of the second column of $\boldsymbol{R}'$ and second row of $\boldsymbol{X}_2$ have been changed. Many operations can be saved in the different steps of calculation of the QR-decomposition, if we learn how to select an optimal path in the heap transform. The optimality relates to minimization of the computation complexity of the QR-decomposition.

Optimal Path of the DsiHT

The path of the heap transformation is an important characteristic of the transformation. In the described above method of QR-decomposition of the matrix by the heap transformations, the ordinary path and path for the strong heap transforms were used. For effective decomposition of non-singular matrices, other well-considered paths P may lead to effective matrix decomposition. The finding such optimal paths is therefore desirable. To illustrate the importance of the path in the heap transformation, we consider the following example with the seven-dimensional vector-generator $x = (1,1,\cdots,1)'$. The heap transformation with this generator is defined by the following six steps of calculations (or Givens rotations):

$$\begin{aligned}
&\text{step } 1: (1,1,1,1,1,1,1) \to \left(\sqrt{2},0,1,1,1,1,1\right) \\
&\text{step } 2: \left(\sqrt{2},0,1,1,1,1,1\right) \to \left(\sqrt{3},0,0,1,1,1,1\right) \\
&\text{step } 3: \left(\sqrt{3},0,0,1,1,1,1\right) \to \left(\sqrt{4},0,0,0,1,1,1\right) \\
&\text{step } 4: \left(\sqrt{4},0,0,0,1,1,1\right) \to \left(\sqrt{5},0,0,0,0,1,1\right) \\
&\text{step } 5: \left(\sqrt{5},0,0,0,0,1,1\right) \to \left(\sqrt{6},0,0,0,0,0,1\right) \\
&\text{step } 6: \left(\sqrt{6},0,0,0,0,0,1\right) \to \left(\sqrt{7},0,0,0,0,0,0\right).
\end{aligned}$$

The calculations require six different square roots. Each component of the vector, except the first one, is processed only ones. We now consider another path P_1 in the heap transformation, which is shown in **Figure 5** where the first column corresponds to the generator.

In this path, the transformation rounds three times the components of the input and intermediate data, and on each round a certain number of heaps are calculated, which then are transferred to one heap. During the first round, three different pairs of components (x_0, x_6), (x_1, x_5), and (x_2, x_4) are processed separately and three heaps of value $\sqrt{2}$ each are calculated, as shown in the second column. Component x_3 is not used in this stage since the dimension of the input is 7, odd number. In the second round, the first three obtained outputs together with x_3 are used to calculate two heaps with values of $\sqrt{3}$ and $\sqrt{4}$. In the last round, these two heaps are rotated to collect the whole energy $\sqrt{7}$ in one heap. The calculation of the heap transform by the path P_1 is thus described by six Givens rotations as follows:

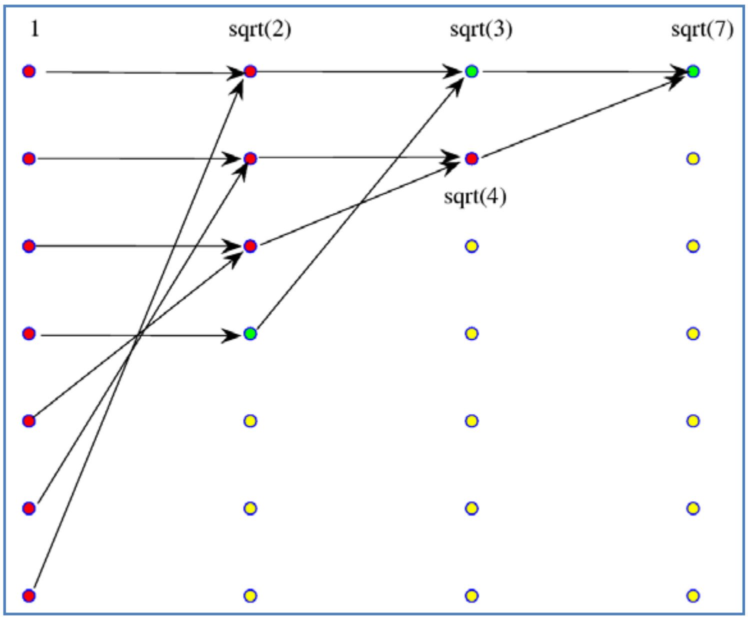


Figure 5. Signal-flow graph of the seven-point heap transformation H.

$$\text{step } 1:\left(\underline{1},1,1,1,1,1,\underline{1}\right)\to\left(\sqrt{2},1,1,1,1,1,0\right)$$
$$\text{step } 2:\left(\sqrt{2},\underline{1},1,1,1,\underline{1},0\right)\to\left(\sqrt{2},\sqrt{2},1,1,1,0,0\right)$$
$$\text{step } 3:\left(\sqrt{2},\sqrt{2},\underline{1},1,\underline{1},0,0\right)\to\left(\sqrt{2},\sqrt{2},\sqrt{2},1,0,0,0\right)$$
$$\text{step } 4:\left(\underline{\sqrt{2}},\sqrt{2},\sqrt{2},\underline{1},0,0,0\right)\to\left(\sqrt{3},\sqrt{2},\sqrt{2},0,0,0,0\right)$$
$$\text{step } 5:\left(\sqrt{3},\underline{\sqrt{2}},\underline{\sqrt{2}},0,0,0,0\right)\to\left(\sqrt{3},\sqrt{4},0,0,0,0,0\right)$$
$$\text{step } 6:\left(\underline{\sqrt{3}},\underline{\sqrt{4}},0,0,0,0,0\right)\to\left(\sqrt{7},0,0,0,0,0,0\right).$$

The underlined numbers in the input vectors show the pairs which are used in rotations in each step of calculation. The same number, six, of steps are used to accomplish the heap transform $H:(1,1,\cdots,1)'\to(7,0,\cdots,0)'$, however there are only four different square roots are calculated, $\sqrt{2},\sqrt{3},\sqrt{4}$, and $\sqrt{7}$. We can also define such a path for the general $N\geq 4$ case, by composing first the transformation on the first round as

$$H_{1;N}=DT_{L-1,T-L}\cdots T_{2,N-3}T_{1,N-2}T_{0,N-1}$$

where L is the integer part of $N/2$ and D is the empty operator if N is even, and the identity operator if N is odd. The transformations $T_{k,n}$ are the Givens rotations of two components of a vector with numbers k and m, when $k,m=0:(N-1)$. On the next round, the transform over the first half of outputs is composed similarly, $H_2=H_{1;N-L}\oplus I_L$. Continuing this process of transformation until the last round which is completed by the transformation $H_r=H_{1;2}\oplus I_{N-2}=T_{1,2}$, we obtain the following heap transformation: $H_{P_1}=H_r\cdots H_2H_1$. The number r of rounds in the path P_1 equals the closest integer to $\log_2(N)$.

Example 4: Let $\boldsymbol{x}$ be the vector $(1,2,3,4)'$. The signal-flow graph of the heap transformation generated by $\boldsymbol{x}$ with path P_1 is given in **Figure 6**.

The orthogonal matrix of this transformation equals

$$\boldsymbol{H}_{P_1}=\begin{bmatrix}0.1826 & 0.3651 & 0.5477 & 0.7303\\ -0.1597 & 0.4176 & 0.6263 & -0.6386\\ 0 & -0.8321 & 0.5547 & 0\\ -0.9701 & 0 & 0 & 0.2425\end{bmatrix}.$$

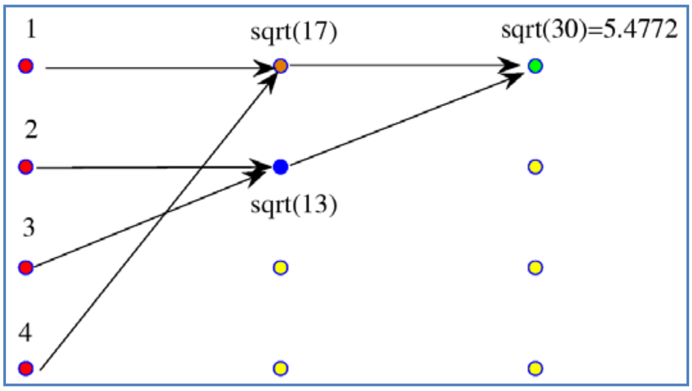


Figure 6. Signal-flow graph of the four-point heap transformation H by the path P_1.

For comparison, we consider the matrix of the heap transformation defined by the ordinary path P,

$$\boldsymbol{H}_P=\begin{bmatrix}0.1826 & 0.3651 & 0.5477 & 0.7303\\ -0.8944 & 0.4472 & 0 & 0\\ -0.3586 & -0.7171 & 0.5976 & 0\\ -0.1952 & -0.3904 & -0.5855 & 0.6831\end{bmatrix}$$

Both transformations result in the same vector, $\boldsymbol{H}_{P_1}[\boldsymbol{x}]=\boldsymbol{H}_P[\boldsymbol{x}]=(5.4772,0,0,0)'$. The matrix $\boldsymbol{H}_{P_1}$ has four zero coefficients which are located symmetrically, and $\boldsymbol{H}_P$ has three zeros composing a triangle.

Example 5: Let $\boldsymbol{x}$ be the vector $(1,1,1,1,1,1,1)'$. The signal-flow graph of the heap transformation generated by $\boldsymbol{x}$ is given in **Figure 5**. The orthogonal matrix of this transformation equals

$$\boldsymbol{H}_{P_1}=\begin{bmatrix}0.3780 & 0.3780 & 0.3780 & 0.3780 & 0.3780 & 0.3780 & 0.3780\\ -0.4364 & 0.3273 & 0.3273 & -0.4364 & 0.3273 & 0.3273 & -0.4364\\ 0 & -0.5000 & 0.5000 & 0 & 0.5000 & -0.5000 & 0\\ -0.4082 & 0 & 0 & 0.8165 & 0 & 0 & -0.4082\\ 0 & 0 & -0.7071 & 0 & 0.7071 & 0 & 0\\ 0 & -0.7071 & 0 & 0 & 0 & 0.7071 & 0\\ -0.7071 & 0 & 0 & 0 & 0 & 0 & 0.7071\end{bmatrix}.$$

This matrix is symmetric with respect to the middle column, and it has one triangle in the center, which is fully filled by zeros. We also can see two not fully filled by zeros triangles in both sides of the middle column. There are total 22 zeros in these triangles, more than in the matrix of the heap transformation defined by the path *P*. Indeed, the matrix of this transformation equals

$$\boldsymbol{H}_P=\begin{bmatrix}0.3780 & 0.3780 & 0.3780 & 0.3780 & 0.3780 & 0.3780 & 0.3780\\ -0.7071 & 0.7071 & 0 & 0 & 0 & 0 & 0\\ -0.4082 & -0.4082 & 0.8165 & 0 & 0 & 0 & 0\\ -0.2887 & -0.2887 & -0.2887 & 0.8660 & 0 & 0 & 0\\ -0.2236 & -0.2236 & -0.2236 & -0.2236 & 0.8944 & 0 & 0\\ -0.1826 & -0.1826 & -0.1826 & -0.1826 & -0.1826 & 0.9129 & 0\\ -0.1543 & -0.1543 & -0.1543 & -0.1543 & -0.1543 & -0.1543 & 0.9258\end{bmatrix}$$

with 15 zero coefficients. Since the matrix $\boldsymbol{H}_{P_1}$ is sparse the application of the heap transformations with the path P_1 leads to the effective QR-decomposition of non-singular matrices. Indeed, in each step of the decomposition, the use of the heap transformation by the path P_1 requires fewer operations, when compared with the heap transformation by the path P, for large dimensions $N>3$. The matrix of the first transformation is sparser than the second one. For instance, in the (5 × 5) example, total 14 multiplications can be saved when using the paths P_1 in the QR-decomposition of the matrix $\boldsymbol{X}=\boldsymbol{U}'\boldsymbol{T}$ instead of P path.

7. Complex DsiHT

In this section, we describe briefly the case when heap transforms are generated by complex vectors and applied on complex vectors. For that we stand on Equation (7) which allows for defining the complex version of the heap transformation. Indeed, we can apply these equations wherein the correlation data are calculated as follows:

$$E_k^2(\boldsymbol{x})=x^2=x_0^2+x_1^2+\cdots+x_{k-1}^2,\ \ E_k(\boldsymbol{z},\boldsymbol{x})=z_0\bar{x}_0+z_1\bar{x}_1+\cdots+z_{k-1}\bar{x}_{k-1},\ \ k=0:(N-1).$$

Here, the bar denotes the operation of the complex conjugate. The transformation defined by these equations is unitary and is called *the complex heap transformation*. The basic two-dimensional transformation *T*, which is defined by a complex vector $(x_0,x_1)'$, and then, is applied to a complex input $(z_0,z_1)'$ is calculated in matrix form as follows:

$$\boldsymbol{T}=\begin{bmatrix} z_1 \\ z_0 \end{bmatrix}\rightarrow\frac{\operatorname{sgn}\left(\operatorname{Real}(x_0)\right)}{\sqrt{|x_0|^2+|x_1|^2}}\begin{bmatrix} \bar{x}_0 & \bar{x}_1 \\ x_1 & x_0 \end{bmatrix}\begin{bmatrix} z_1 \\ z_0 \end{bmatrix}.$$

When $(z_0,z_1)'=(x_0,x_1)'$, we obtain the real transform $\boldsymbol{T}\cdot(x_0,x_1)'=\left(\operatorname{sign}\left(\operatorname{Real}(x_0)\right)\sqrt{|x_0|^2+|x_1|^2},0\right)'$. It should be noted that this transformation differs from the known definition of the complex Givens rotation [4], whose matrix is calculated as

$$\boldsymbol{R}(c,s)=\begin{bmatrix} c & s \\ -\bar{s} & c \end{bmatrix}=\frac{1}{\sqrt{|x_0|^2+|x_1|^2}}\begin{bmatrix} |x_0| & \operatorname{sgn}(x_0)\bar{x}_1 \\ -\overline{\operatorname{sgn}(x_0)}x_1 & |x_0| \end{bmatrix}$$

where the complex sign function is defined by $\operatorname{sgn}(x_0)=x_0/|x_0|$. This rotation results in the following complex transform $\boldsymbol{R}(c,s)\cdot(x_0,x_1)'=\left(B\sqrt{|x_0|^2+|x_1|^2},0\right)'$ where $B=\operatorname{sign}(x_0)$ is a complex number with norm one.

Example 6: Let $N=4$ and let the complex vector $\boldsymbol{x}$ to be $\boldsymbol{x}=(1+3\mathrm{i},2+\mathrm{i},3+4\mathrm{i},4-2\mathrm{i})'$. The complex heap transformation generated by this vector has the following matrix:

$$\boldsymbol{T}_x=\begin{bmatrix} 0.1291-0.3873\mathrm{i} & 0.2582-0.1291\mathrm{i} & 0.3873-0.5164\mathrm{i} & 0.5164+0.2582\mathrm{i} \\ -0.4082+0.4082\mathrm{i} & 0.8165 & 0 & 0 \\ -0.6124+0.2041\mathrm{i} & -0.4082-0.2041\mathrm{i} & 0.6124 & 0 \\ 0.0408+0.2858\mathrm{i} & -0.1225+0.1633\mathrm{i} & -0.0816+0.4491\mathrm{i} & 0.8165 \end{bmatrix}.$$

The transform of the vector-generator equals $\boldsymbol{T}_x^*[\boldsymbol{x}]=(7.7460,0,0,0)'$, where $7.7460=\|x\|$. The matrix $\boldsymbol{T}_x^*$ is the complex conjugate to $\boldsymbol{T}_x$ and its first row is the normalized generator, *i.e.*, $(1+3\mathrm{i},2+\mathrm{i},3+4\mathrm{i},4-2\mathrm{i})/7.7460$. The complex heap transformations can be used for triangularization of the square complex matrix in a way similar to the real case described above. As an example, we consider the following matrix:

$$\boldsymbol{X}=\begin{bmatrix} 1+\mathrm{i} & 2-3\mathrm{i} & 3+4\mathrm{i} \\ 2-3\mathrm{i} & 3+2\mathrm{i} & 2-2\mathrm{i} \\ 3-\mathrm{i} & 4+3\mathrm{i} & 4-2\mathrm{i} \end{bmatrix}.$$

The method of heap transforms results in the following decomposition of $\boldsymbol{X}=\boldsymbol{R}'\boldsymbol{X}_2$ where $\boldsymbol{R}$ is the unitary matrix

$$\boldsymbol{R}=\begin{bmatrix} 0.2000-0.2000\mathrm{i} & 0.4000+0.6000\mathrm{i} & 0.6000+0.2000\mathrm{i} \\ 0.4329+0.7837\mathrm{i} & -0.0299-0.1493\mathrm{i} & 0.3583-0.2165\mathrm{i} \\ 0.2612-0.2239\mathrm{i} & 0.0746-0.6718\mathrm{i} & 0.2612+0.5971\mathrm{i} \end{bmatrix}$$

and $\boldsymbol{X}_2$ is the upper triangular matrix

$$\boldsymbol{X}_2=\begin{bmatrix} 5 & 2.2000+3.8000\mathrm{i} & 6.2000+0.2000\mathrm{i} \\ 0 & 5.3591 & -1.1942+2.2616\mathrm{i} \\ 0 & 0 & 2.7243+0.7464\mathrm{i} \end{bmatrix}.$$

8. Conclusion

A novel decomposition of nonsingular real and complex matrices by fast heap transforms has been described, when the paths P of the transforms are natural. The path of the heap transform is an important characteristic of the transform, and other paths P can be found, that may result in more sparse matrices than the path P does. The application of such paths will lead to effective matrix decomposition, as shown on examples with the strong heap transform. We have considered only the case of nonsingular matrices of square size. However, the calculations presented above can be applied to the case of non square matrices, as well.

References

[1] Horn, R.A. and Charles, R.J. (1985) Matrix Analysis. Cambridge University Press, Cambridge. http://dx.doi.org/10.1017/CBO9780511810817

[2] Moon, T.K. and Stirling, W.C. (2000) Mathematical Methods and Algorithms for Signal Processing. Prentice Hall, Upper Saddle River.

[3] Morrison, D.D. (1960) Remarks on the Unitary Triangularization of a Nonsymmetric Matrix. *Journal ACM*, **7**, 185-186. http://dx.doi.org/10.1145/321021.321030

[4] Golub, G.H. and Van Loan, C.F. (1996) Matrix Computations. 3rd Edition, Johns Hopkins, Balimore.

[5] Householder, A.S. (1958) Unitary Triangulation of a Nonsymmetric Matrix. *Journal ACM*, **5**, 339-342. http://dx.doi.org/10.1145/320941.320947

[6] Anderson, E., Bai, Z., Bischof, C.H., *et al.* (1999) LAPACK Users' Guide. 3rd Edition, SIAM Philadelphia, Philadelphia. http://dx.doi.org/10.1137/1.9780898719604

[7] Bindel, D., Demmel J., Kahan, W. and Marques, O. (2001) On Computing Givens Rotations Reliably and Efficiently. LAPACK Working Note 148, University of Tennessee, UT-CS-00-449.

[8] Demmel, J., Grigori, L., Hoemmen, M. and Langou, J. (2012) Communication-Optimal Parallel and Sequential QR and LU Factorizations. *SIAM Journal on Scientific Computing*, **34**, 206-239. http://dx.doi.org/10.1137/080731992

[9] Grigoryan, A.M. and Grigoryan, M.M. (2006) Nonlinear Approach of Construction of Fast Unitary Transforms. In: *Proceedings of the* 40*th Annual Conference on Information Sciences and Systems* (*CISS* 2006), Princeton University, Princeton, 1073-1078. http://dx.doi.org/10.1109/CISS.2006.286625

[10] Grigoryan, A.M. and Grigoryan, M.M. (2007) Discrete Unitary Transforms Generated by Moving Waves. *Proceedings of the International Conference*: *Wavelets XII*, *SPIE*: *Optics* + *Photonics* 2007, San Diego.

[11] Grigoryan, A.M. and Grigoryan, M.M. (2007) New Discrete Unitary Haar-Type Heap Transforms. *Proceedings of the International Conference: Wavelets XII*, *SPIE: Optics + Photonics*, **6701**.

[12] Grigoryan, A.M. and Grigoryan, M.M. (2009) Brief Notes in Advanced DSP: Fourier Analysis with MATLAB. CRC Press, Taylor and Francis Group, Philadelphia.