Exponential Decomposition Using Displacement Structure

Sanzheng Qiao[a] and Jun Zhu[b]

[a]Department of Computing and Software
McMaster University
Hamilton, Ontario L8S 4L7 Canada

[b]Department of Electrical and Computer Engineering
McMaster University
Hamilton, Ontario L8S 4L7 Canada

ABSTRACT

In this paper, we present a novel approach to the problem of exponential decomposition. This method can compute the knots in $O(n^2)$ floating-point operations and $O(n)$ storage, where n is the length of the signal sequence.

Keywords: Exponential decomposition, Hankel matrix, Displacement structure, Complex-orthogonal transformations

1. INTRODUCTION

The problem of decomposing a signal into a sum of complex exponential sequences arises widely in signal and image processing applications. This problem can be applied as a modeling tool[10] and can also be used indirectly in areas such as speech processing,[5] coding theory[1] and biomedical signal processing.[11]

The exponential decomposition problem can be defined as follows:
given a finite sequence

$$s = (s_1, s_2, \cdots, s_n)^{\mathrm{T}},$$

we want to determine a minimal integer r, complex coefficients a_i, and distinct complex knots z_i $(i = 1, \cdots, r)$ that satisfy the following signal model:

$$s_k = \sum_{i=1}^{r} a_i z_i^{k-1}. \tag{1}$$

Once the knots z_i are found, the coefficient vector $a = (a_1, a_2, \ldots, a_r)^{\mathrm{T}}$ can be determined by solving the following Vandermonde system:

$$s = Wa, \qquad W = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ z_1 & z_2 & \cdots & z_r \\ z_1^2 & z_2^2 & \cdots & z_r^2 \\ \vdots & \vdots & \vdots & \vdots \\ z_1^{r-1} & z_2^{r-1} & \cdots & z_r^{r-1} \end{bmatrix}. \tag{2}$$

A Hankel matrix is a matrix with constant elements along its antidiagonals. For example, suppose s_i are signals satisfying the model (1) and suppose $k + r \leq n$, we construct a signal Hankel matrix:

$$H = \begin{bmatrix} s_1 & s_2 & \cdots & s_r \\ s_2 & s_3 & \cdots & s_{r+1} \\ s_3 & s_4 & \cdots & s_{r+2} \\ \vdots & \vdots & \cdots & \vdots \\ s_k & s_{k+1} & \cdots & s_{k+r-1} \\ s_{k+1} & s_{k+2} & \cdots & s_{k+r} \end{bmatrix}. \tag{3}$$

Sanzheng Qiao: Email: qiao@cas.mcmaster.ca; Telephone: (905)525-9140 ext. 27234; Fax: (905)524-0340

All previously known linear algebraic exponential decomposition methods, such as Prony,[9] Kung's Hankel SVD,[6] and implicit Lanczos process,[12] can be unified within a single matrix pencil based framework.[12] This paper will contribute another fast algorithm for solving the exponential decomposition. First, this method, using a triangular decomposition of the signal Hankel matrix in (3), computes a tridiagonal matrix whose eigenvalues are the knots z_i in the exponential decomposition. Using displacement structure of Hankel matrices, we are able to compute the tridiagonal matrix in $O(n^2)$ floating-point operations and $O(n)$ storage. Second, complex-orthogonal transformations are applied to the tridiagonal matrix to obtain the eigenvalues, i.e., the knots.

This paper is organized as follows. After describing the matrix pencil framework and demonstrating how this new method fits in the framework, we give general steps in this new algorithm in §2. We devote §3 to the triangularization of the signal Hankel matrix using displacement structure, which is the key step in this new approach. In §4, we present some numerical examples.

In this paper, we adopt Matlab colon-notations for matrix and vector subsets. For example, $A_{i:j,k:l}$ denotes the submatrix of A containing rows i through j, and columns k through l, and $A_{:,k:l}$ represents the submatrix of A consisting of columns k through l.

2. MATRIX PENCIL FRAMEWORK

Vandevoorde[12] presented a matrix pencil framework which unifies previously known exponential decomposition methods. In general, for any nonsingular F and G, the knots z_i are the eigenvalues of the pencil-equation

$$FH_2Gv - \lambda FH_1Gv = 0, \tag{4}$$

where

$$H_1 = H_{1:k,:} \quad \text{and} \quad H_2 = H_{2:k+1,:}$$

are two k-by-r submatrices of the Hankel matrix in (3).

Assuming that $k = r$, we first expand H into a $2r \times 2r$ Hankel matrix:

$$\hat{H} = \begin{bmatrix} s_1 & \cdots & s_{2r} \\ \vdots & \iddots & \\ s_{2r} & & 0 \end{bmatrix}. \tag{5}$$

We also assume that $\hat{H}$ is strongly nonsingular, i.e., all its leading principal submatrices are nonsingular. Suppose that we have a triangular decomposition $\hat{H} = LDL^{\mathrm{T}}$, where L is unit lower triangular and $D = \mathrm{diag}(d_1, \cdots, d_{2r})$ diagonal. Since the Hankel matrix H in (3) equals $\hat{H}_{1:k+1,1:r}$, we have

$$H = L_{1:k+1,1:r}D_{1:r,1:r}L^{\mathrm{T}}_{1:r,1:r}$$

and

$$H_1 \equiv H_{1:k,:} = L_{1:k,1:r}D_{1:r,1:r}L^{\mathrm{T}}_{1:r,1:r} \quad \text{and} \quad H_2 \equiv H_{2:k+1,:} = L_{2:k+1,1:r}D_{1:r,1:r}L^{\mathrm{T}}_{1:r,1:r}.$$

Let $F = I$, $G = (L_{1:r,1:r})^{-\mathrm{T}}(D_{1:r,1:r})^{-1}$ in the framework (4), the eigenvalues of a matrix Y satisfying

$$L_{1:k,1:r}Y = L_{2:k+1,1:r} \tag{6}$$

are the desired knots. Using Lanczos process,[8] we can show that there is a tridiagonal matrix T satisfying (6), that is

$$L_{1:k,1:r}T^{\mathrm{T}} = L_{2:k+1,1:r}.$$

Thus, the eigenvalues of T^{T} are the knots. The tridiagonal T is of the form

$$T = \begin{bmatrix} \alpha_1 & \beta_1^2 & & & 0 \\ 1 & \alpha_2 & \beta_2^2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & \alpha_{r-1} & \beta_{r-1}^2 \\ 0 & & & 1 & \alpha_r \end{bmatrix}. \tag{7}$$

Suppose that we already know how to get two consecutive columns of L, which we will show in §3, then the following recurrence gives us α and β of T.

ALGORITHM 1. *(tridiagonalization)*

Compute $\alpha_i = L_{i+1,i} - L_{i,i-1}$;

$\beta_i^2 = L_{i+2,i} - \alpha_i L_{i+1,i} - L_{i+1,i-1}$;

The tridiagonal matrix T can be transformed into a complex symmetric tridiagonal matrix by scaling its rows and columns. We then apply the complex-orthogonal transformations[8] in the QR method[3] to compute its eigenvalues, i.e., the knots.

3. TRIANGULARIZATION USING DISPLACEMENT STRUCTURE

In this section, we describe the steps in finding a triangular decomposition $H = LDL^{\mathrm{T}}$. First, we establish a relation between triangularization and Schur reduction. Second, we connect Schur reduction with displacement since Hankel displacement has low rank. Third, we describe a fast Schur reduction using generators of Hankel displacement. Finally, we show how to obtain the tridiagonal matrix T in (7) efficiently.

3.1. General Schur reduction

Suppose that we partition $H \in C^{n\times n}$

$$H \equiv \begin{bmatrix} A & B \\ B^{\mathrm{T}} & D \end{bmatrix}$$

where $A \in C^{i\times i}$ and let $S \in C^{n\times n}$

$$S \equiv \begin{bmatrix} 0 & 0 \\ 0 & \bar{S} \end{bmatrix}.$$

If A is nonsingular, the $(n-i)\times(n-i)$ submatrix $\bar{S} \equiv D - B^{\mathrm{T}}A^{-1}B$ is called the Schur complement of A. The following recursive computation of the Schur complements S_i of the leading principal submatrices of a given matrix H leads us to the triangular factorization procedure.[3]

Let

$$H_0 \equiv S_0 \equiv H \in C^{n\times n},$$

$L = [l_1, l_2, \cdots, l_n]$ be unit lower triangular, $U = [u_1, u_2, \cdots, u_n]^{\mathrm{T}} = L^{\mathrm{T}}$ unit upper triangular, $H = LDU = \sum_{i=1}^{n} l_i d_i u_i^{\mathrm{T}}$, we define the *ith reduced matrix* as

$$H_i \equiv \begin{bmatrix} 0_{i\times i} & 0_{i\times(n-i)} \\ 0_{(n-i)\times i} & S_i \end{bmatrix}.$$

Then we have

$$H_{i-1} = l_i d_i u_i^{\mathrm{T}} + H_i. \tag{8}$$

Given H_{i-1}, recalling that L is unit lower triangular and U is unit upper triangular, we can determine l_i, d_i, and u_i in (8) by

$$d_i l_i = H_{i-1} e_i, \quad \text{and} \quad d_i u_i = H_{i-1}^{\mathrm{T}} e_i, \tag{9}$$

where e_i is the unit vector with one at the ith position and zeros elsewhere. The computation of the reduced matrix H_i from H_{i-1} is called (one step) *Schur reduction*.[3] Using r Schur-reduction steps, we can obtain the (r-step) partial triangular factorization,

$$H = \sum_{i=1}^{r} l_i d_i u_i^{\mathrm{T}} + H_r \equiv L_{:,1:r} D_{1:r,1:r} (L_{:,1:r})^{\mathrm{T}} + H_r.$$

3.2. Schur reduction using Hankel displacement

Although Schur reduction is related to triangularization, it does not retain Hankel structure. To obtain efficient solution, we use Hankel displacement.[2] Let

$$\Delta H \equiv Z_n H - H Z_n^{\mathrm{T}}, \quad H \in C^{n\times n}, \tag{10}$$

where Z_n is the $n \times n$ lower shift matrix with ones on the first subdiagonal and zeros everywhere else. The matrix ΔH is called the *Hankel displacement* of H with respect to the *displacement operators* Z_n. The rank of ΔH is called the *Hankel displacement rank* of H.

We can see that if

$$H = \begin{bmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_1 & \iddots & \iddots & h_n \\ \vdots & \iddots & \iddots & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-2} \end{bmatrix},$$

then the displacement

$$\Delta H = Z_n H - H Z_n^{\mathrm{T}} = \begin{bmatrix} 0 & -h_0 & -h_1 & \cdots & -h_{n-1} \\ h_0 & & & & \\ h_1 & & & & \\ \vdots & & & 0 & \\ h_{n-1} & & & & \end{bmatrix}$$

has rank 2.

A displacement of a matrix H can characterize the matrix H if we can solve the equation (10) for H uniquely.[2] To achieve the uniqueness, we extend $H \in C^{n\times n}$ to a larger $\bar{H} \in C^{2n\times 2n}$, such that H is a leading principal submatrix of $\bar{H}$, i.e.,

$$\bar{H} = \begin{bmatrix} H & U_n \\ U_n & 0 \end{bmatrix} \in C^{2n\times 2n}, \tag{11}$$

where U_n is the "reverse upper triangular" Hankel matrix with zero elements in the lower right corner such that $\bar{H}$ in (11) is Hankel. Now we can find l_i uniquely by working on $\bar{H}$ instead of H itself. In the following, we assume the given Hankel matrix H is of the form as $\bar{H}$. Also, we consider strongly nonsingular matrices for triangularization.

Now we apply displacement to the reduced matrices H_i. The displacement ranks of the reduced matrices do not increase,[2] i.e.

$$\mathrm{rank}(\Delta H_i) \le \mathrm{rank}(\Delta H_{i-1}), \quad (0 \le i \le r).$$

In the following, we show that by successively computing ΔH_i rather than H_i, the Schur reduction procedure in (8) can be done efficiently. Recalling the following definition of displacement

$$\Delta H_{i-1} = Z_n H_{i-1} - H_{i-1} Z_n^{\mathrm{T}},$$

from (9), we get

$$(\Delta H_{i-1}) e_i = d_i Z_n l_i \quad \text{and} \quad (\Delta H_{i-1})^{\mathrm{T}} e_i = -d_i Z_n u_i. \tag{12}$$

Therefore, we can compute l_i, d_i, and u_i from ΔH_{i-1} as the following:

$$d_i l_i = Z_n^{\mathrm{T}} (\Delta H_{i-1}) e_i \quad \text{and} \quad d_i u_i = -Z_n^{\mathrm{T}} (\Delta H_{i-1})^{\mathrm{T}} e_i. \tag{13}$$

From (8), we obtain an update scheme of the displacement of the reduced matrix H_i:

$$\begin{aligned} \Delta H_i &= Z_n H_i - H_i Z_n^{\mathrm{T}} \\ &= \Delta H_{i-1} - Z_n l_i d_i u_i^{\mathrm{T}} + l_i d_i u_i^{\mathrm{T}} Z_n^{\mathrm{T}} \end{aligned} \tag{14}$$

3.3. Fast Schur reduction using Hankel generators

Equation (14) shows that ΔH_i is a rank-2 update of ΔH_{i-1}. This is undesirable, since each update requires a rank-2 modification. By utilizing a generator of a Hankel matrix,[2] we can improve efficiency .

Since ΔH is of rank 2, for a given Hankel matrix $H \in C^{n\times n}$, we can find a matrix pair $\{X, Y\}$ such that

$$\Delta H = XY^T, \quad X \equiv [x_1, x_2] \in C^{n\times 2}, \quad Y \equiv [y_1, y_2] \in C^{n\times 2}.$$

Such matrix pair is called a Hankel generator of H with respect to Z_n. The length of the generator with respect to Z_n is 2. A generator with respect to Z_n with its length equal to the displacement rank is called a minimal generator.[2]

It is easy to verify

$$\Delta H = \begin{bmatrix} 0 & -h_0 & \cdots & -h_{n-2} \\ h_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ h_{n-2} & 0 & \cdots & 0 \end{bmatrix} = XY^{\mathrm{T}}$$

where

$$X = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & h_0 & \cdots & h_{n-2} \end{bmatrix}^{\mathrm{T}}, \quad Y = \begin{bmatrix} 0 & -h_0 & \cdots & -h_{n-2} \\ 1 & 0 & \cdots & 0 \end{bmatrix}^{\mathrm{T}}. \tag{15}$$

By using the generators of the displacement of Hankel matrices, we can obtain a fast algorithm for the Schur reduction. Let

$$\Delta H_{i-1} = X^{(i-1)} Y^{(i-1)\mathrm{T}} \quad \text{with} \quad \Delta H_0 = \Delta H,\ X^{(0)} = X, \text{ and } Y^{(0)} = Y,$$

then, from (14), the generator of H_i can be obtained by that of H_{i-1} and l_i, d_i, and u_i using

$$X^{(i)} = [X^{(i-1)}, -Z_n d_i l_i, d_i l_i], \quad Y^{(i)} = [Y^{(i-1)}, u_i, Z_n u_i]. \tag{16}$$

Since the displacement operator, Z_n, is simple, the matrix-vector multiplications in (16) are trivial. Therefore, the triangular factorization with displacement operator using the above reduction is faster than the original reduction (14).

3.4. Triangular factorization of Hankel matrices

Equation (16) shows that the number of columns in $X^{(i)}$ and $Y^{(i)}$ grows by two for each update. By introducing the concept of proper generator,[2] we can maintain the number of columns in $X^{(i)}$ and $Y^{(i)}$ to two.

A Hankel generator $\{X^{(i-1)}, Y^{(i-1)}\}$ of H_{i-1} with respect to Z_n is called *proper* if all the elements in the ith row of $X^{(i-1)}$ and above, except for the element $X^{(i-1)}_{i,1}$, are zero, and all elements in the ith row of $Y^{(i-1)}$ and above, except the element $Y^{(i-1)}_{i,2}$, are zero. It is obvious that the generator $\{X, Y\}$ in (15) is proper. The use of proper generator leads to the most efficient method for the Schur reduction.

Let $\{X^{(i-1)},\ Y^{(i-1)}\}$ be proper, then equation (13) implies

$$\begin{aligned} d_i l_i &= Z_n^{\mathrm{T}} (X^{(i-1)} Y^{(i-1)\mathrm{T}}) e_i = Y^{(i-1)}_{i,2} Z_n^{\mathrm{T}} X^{(i-1)}_{:,2} \\ d_i u_i &= -Z_n^{\mathrm{T}} (Y^{(i-1)} X^{(i-1)\mathrm{T}}) e_i = -X^{(i-1)}_{i,1} Z_n^{\mathrm{T}} Y^{(i-1)}_{:,1} \end{aligned} \tag{17}$$

Therefore, (16) becomes

$$\begin{aligned} X^{(i)} &= [X^{(i-1)},\ -Y^{(i-1)}_{i,2} X^{(i-1)}_{:,2},\ d_i l_i] \\ Y^{(i)} &= [Y^{(i-1)},\ u_i,\ -d_i^{-1} X^{(i-1)}_{i,1} Y^{(i-1)}_{:,1}], \end{aligned}$$

which shows that

$$X^{(i)} Y^{(i)\mathrm{T}} = (X^{(i-1)}_{:,1} - X^{(i-1)}_{i,1} l_i)(Y^{(i-1)}_{:,1})^{\mathrm{T}} + X^{(i-1)}_{:,2}((Y^{(i-1)}_{:,2})^{\mathrm{T}} - Y^{(i-1)}_{i,2} u_i^{\mathrm{T}}).$$

It follows that we can choose

$$X^{(i)} = [(X_{:,1}^{(i-1)} - X_{i,1}^{(i-1)} l_i), \quad X_{:,2}^{(i-1)}] \quad \text{and} \quad Y^{(i)} = [Y_{:,1}^{(i-1)}, \quad (Y_{:,2}^{(i-1)} - Y_{i,2}^{(i-1)} u_i)]. \tag{18}$$

Thus we can maintain the number of columns in $X^{(i)}$ and $Y^{(i)}$ to two. However, they are not necessarily proper. To make $X^{(i)}$ and $Y^{(i)}$ proper, we eliminate $X_{i,2}^{(i)}$ and $Y_{i,1}^{(i)}$. Using the elimination matrix

$$M = \begin{bmatrix} 1 & \mu \\ 0 & 1 \end{bmatrix}$$

where $\mu = -X_{i,2}^{(i)} / X_{i,1}^{(i)}$, we have

$$X^{(i)} M = \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ * & 0 \\ * & * \\ \vdots & \vdots \\ * & * \end{bmatrix},$$

which is proper. It is easy to verify that

$$M^{-1} = \begin{bmatrix} 1 & -\mu \\ 0 & 1 \end{bmatrix}.$$

On the other hand, it is easy to check that X and Y in equation (15) are related by

$$Y = XP^{\mathrm{T}}, \quad \text{where} \quad P = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

We call the generator $\{X, XP^{\mathrm{T}}\}$ skew symmetric. Equation (18) shows that if $\{X^{(i-1)}, Y^{(i-1)}\}$ is skew symmetric, then $\{X^{(i)}, Y^{(i)}\}$ is also skew symmetric. It is also important to notice that $\{X^{(i)}, Y^{(i)}\}$ is still skew symmetric after we make it proper, because

$$\begin{aligned} XY^{\mathrm{T}} &= XPX^{\mathrm{T}} \\ &= (XM)(M^{-1}P)X^{\mathrm{T}} \\ &= (XM)(PM^{\mathrm{T}})X^{\mathrm{T}} \\ &= (XM)P(XM)^{\mathrm{T}}. \end{aligned}$$

With a skew symmetric generator $\{X, XP^{\mathrm{T}}\}$, we only compute $X^{(i)}$ and make it proper. The following algorithm gives the factorization $H = LDL^{\mathrm{T}}$. Also, since L is uint lower triangular, the ith entry of $d_i l_i$ in (17) is d_i. Because $L = U^{\mathrm{T}}$, only L is computed.

ALGORITHM 2. *(r-step partial triangular factorization of Hankel matrix)*

> *for* $i = 1$ *to* r
>
> > *Construct a proper generator* $\{X^{(i-1)},\ X^{(i-1)P^{\mathrm{T}}}\}$ *of* H_{i-1};
> > *Compute* d_i, l_i *using (17);*
> > *Obtain a generator* $\{X^{(i)},\ X^{(i)}P^{\mathrm{T}}\}$ *of* H_i *by (18);*
>
> *end;*
> *return* $L = [l_1, l_2, \cdots, l_r]$;

We can combine the steps in Algorithm 1 with Algorithm 2 to get the final tridiagonal matrix T.

Algorithm 3. *(Tridiagonalization using Displacement Structure)*
Given a signal sequence $s_1, s_2, ..., s_{2r}$ in (5), this algorithm computes α_i and β_i in the tridiagonal T of (7) using displacement structure.

Initialize the generator X of the displacement matrix of H;
Initialize a column $L_{:,pre}$ of the low triangular L to zero;
for $i = 1$ to r
Make the generator X proper;
Compute current column $L_{:,cur}$ using (17);
Compute $\alpha_i = L_{i+1,cur} - L_{i,pre}$;
if $(i < r)$
$\beta_i^2 = L_{i+2,cur} - \alpha_i L_{i+1,cur} - L_{i+1,pre}$;
Update the generator X using (18).
$L_{:,prev} = L_{:,cur}$;
end.
end.

This algorithm requires $O(n^2)$ floating-point computation and $O(n)$ storage.

4. EXAMPLES

The growth rate of the floating-point operation count, shown in Figure 1, indicates that Lanczos method and displacement method are more efficient than the Prony and Kung's HSVD method.

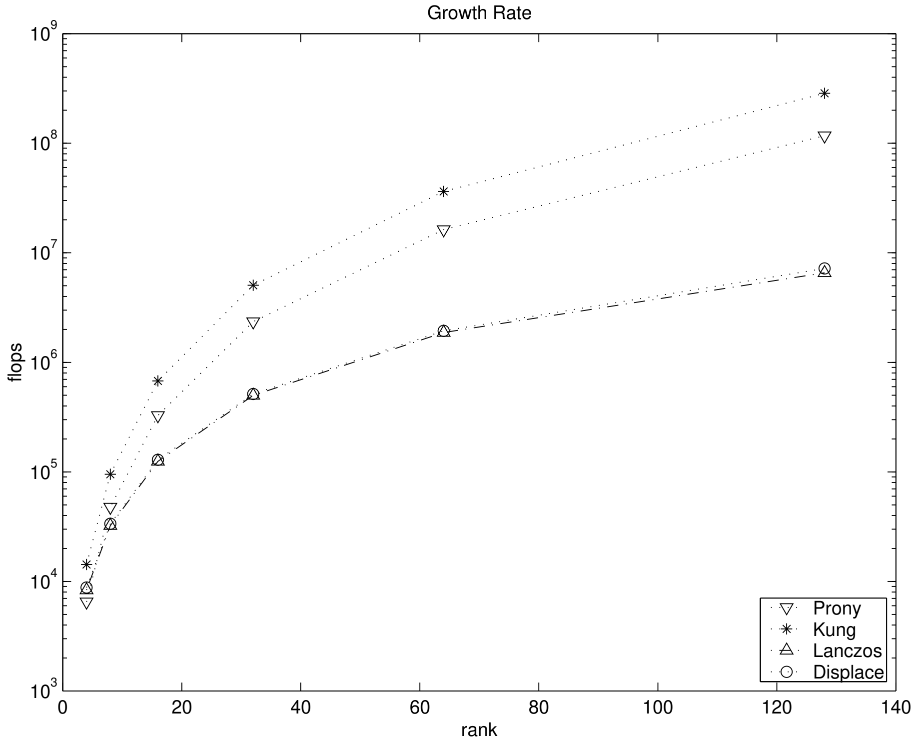

Figure 1. Growth Rate

r	**Prony**	**Kung**	**Lanczos**	**Displace**
4	0.00778	0.00783	0.00778	0.00778
8	0.01302	0.01399	0.01302	0.01302
16	0.00542	0.00539	0.00542	0.00542
32	0.00612	0.00588	0.00612	0.00612
64	0.00254	0.00251	0.00254	0.00254
128	0.00230	0.00250	0.00230	0.00230

Table 1. Comparison of roots difference

Table 1 numerically demonstrates the norm of distance between pairs of the origianl knots and the computed knots. The original knots were generated as r distinct points evenly spaced on a unit circle. The original signals were generated according to the exponential decomposition model with random coefficients and signal to noise ratio 40 db. The computational errors were calculated as the 2-norm of the minimal difference between the original knots and the computed knots by matching the pairs.

REFERENCES

1. D.L.Boley, R.P.Brent, G.H.Golub, and F.T.Luk, *Algorithmic fault tolerance using the Lanczos method*, SIAM J.Matrix Analy. Appl., 13 (1992), pp. 312-332.
2. J. Chun and T. Kailath, *Displacement Structure for Hankel, Vandermonde, and Related (Derived) Matrices*, Linear Algebra and its applications, 151(1991), pp.199-227.
3. Gene H. Golub and Charles F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1990.
4. T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1980.
5. R. Kumaresan and D.W.Tufts, *Estimating the parameters of exponentially damped sinusoids and pole-zero modeling in noise*, IEEE Trans. Acoust. Speech and Sig. Proc., 30 (1982), pp. 833-840.
6. S. Y. Kung, K.S. Arun, and B.D. Rao, *State-space and singular value decomposition-based approximation methods for the harmonic retrieval problem*, J.Opt.Soc.Am., 73(1983), pp.1799-1811.
7. C. Lanczos, Applied Analysis, Prentice-Hall, Englewood Cliffs, 1965.
8. F.T. Luk and Sanzheng Qiao, *Using Complex-Orthogonal Transformation to Diagonalize a Complex Symmetric Matrix*, Advanced Signal Processing: Algorithms, Architectures, and Implementations VII, Proc. SPIE Vol. 3162, pp. 418-425, July 1997.
9. R. Prony, *Essai experimental et anaylytique sur les lois de la delatabilite et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, a dfferentes temperature,* J. de l´ Ecole Polytechnique, 1(1795), pp. 24-76.
10. W.M. Steedly and R.L.Moses, *High resolution exponential modeling of fully polarized radar returns*, IEEE Trans. on Aerospace and Electronic Systems, 27 (1991), pp.459-469.
11. Sabine Van Huffel, *Mathematics in biomedical signal processing*, Fifth IMA International Conference, Dec. 30, 2000 or http://www.esat.kuleuven.ac.be/sista/members/biomed/
12. David Vandevoorde, *A fast exponential decomposition algorithm and its applications to structured matrices,* Ph.D. thesis, Rensselaer Polytechnic Institute, Troy, New York, 1996.