# Using Complex-Orthogonal Transformations to Diagonalize a Complex Symmetric Matrix

Franklin T. Luk
Department of Computer Science, Rensselaer Polytechnic Institute
Troy, New York 12180 USA

Sanzheng Qiao
Department of Computer Science and Systems, McMaster University
Hamilton, Ontario L8S 4K1 CANADA

## ABSTRACT

In this paper, we propose the use of complex-orthogonal transformations for finding the eigenvalues of a complex symmetric matrix. Using these special transformations can significantly reduce computational costs because the tridiagonal structure of a complex symmetric matrix is maintained.

**Keywords:** Complex symmetric matrix, eigenvalue decomposition, complex-orthogonal transformation, tridiagonal matrix.

## 1. INTRODUCTION

In many applications, such as directions of arrival, nuclear magnetic resonance, image restoration, system identification, speech processing, coding theory, and so on, the signals are modeled by linear combinations of a set of exponentials:

$$s_k = \sum_{i=1}^{r} a_i z_i^k, \tag{1.1}$$

where $a_i$ are complex coefficients and $z_i$ are complex exponentials. The problem of the exponential decomposition is as follows. Given a finite sequence of signals:

$$s = \{s_0, s_1, ..., s_n\},$$

find the smallest integer $r$, and sets of $r$ coefficients $\{a_i\}$ and $r$ exponentials $\{z_i\}$ so that (1.1) is satisfied. The exponential decomposition problem has a long history. More than two centuries ago, Prony [6] tackled this problem by finding roots of a polynomial; an English translation of [6] can be found in [7]. Kung et al. [5] presented an approach based on the singular value decomposition to avoid the numerically undesirable procedure of polynomial root-finding. Vandevoorde [7] proposed a class of fast methods that uses Lanczos tridiagonalization as an initial step.

This paper is organized as follows. In §2, we describe a Lanczos tridiagonalization approach to the exponential decomposition problem and show how a complex symmetric tridiagonal eigenproblem arises. In §3, we introduce the complex-orthogonal transformations and discuss their properties and application to the diagonalization of a complex symmetric matrix. Finally, an overall procedure for the exponential decomposition and numerical experimental results are presented in §4 and §5, respectively.

## 2. LANCZOS METHOD

Vandevoorde [7] proposed to use Lanczos tridiagonalization for the exponential decomposition, which leads to a complex symmetric tridiagonal eigenproblem. Given a Krylov matrix:

$$K(e, D_z) \equiv (e, D_z e, ..., D_z^{r-1} e), \tag{2.1}$$

where

$$e \equiv (1, 1, ..., 1)^T$$

and

$$D_z \equiv \mathrm{diag}(z_1, z_2, ..., z_r),$$

the Lanczos method finds a basis $q_1, q_2, ..., q_r$ for a Krylov subspace given by span$\{e, D_z e, ..., D_z^{r-1} e\}$. The matrix $Q \equiv (q_1, q_2, ..., q_r)$ is orthogonal with respect to $D_a \equiv \mathrm{diag}(a_1, a_2, ..., a_r)$, i.e., the matrix $Q^T D_a Q$ is diagonal; $Q$ also satisfies

$$QT = D_z Q, \tag{2.2}$$

where $T$ is a tridiagonal matrix:

$$T = \begin{pmatrix} \alpha_1 & \beta_1^2 & & & 0 \\ 1 & \alpha_2 & \beta_2^2 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & \alpha_{r-1} & \beta_{r-1}^2 \\ 0 & & & 1 & \alpha_r \end{pmatrix}. \tag{2.3}$$

Equation (2.2) implies that the eigenvalues of $T$ are the exponentials $z_1, z_2, ..., z_r$. As a byproduct, the Lanczos method produces a $QR$ decomposition of the Krylov matrix $K(e, D_z)$:

$$K = QR, \tag{2.4}$$

where the matrix $Q$ is $D_a$-orthogonal, $R \equiv (e_1, Te_1, ..., T^{r-1}e_1)$ is upper triangular with a unit diagonal and $e_1 \equiv (1, 0, ..., 0)^T$. From the definition of $R$, we derive a relation between two consecutive columns of $R$:

$$R(:, j+1) = TR(:, j),$$

leading to a recursion on the rows of $R$:

$$\beta_i^2 R(i+1, i : r-2) = R(i, i+1 : r-1) - \alpha_i R(i, i : r-2) - R(i-1, i : r-2).$$

Using $R(i, i) = 1$, $R(i+1, i) = 0$, and the above recursion, we can compute $\alpha_i$, $\beta_i$, and the row $R(i+1, :)$ from the two rows $R(i-1, :)$ and $R(i, :)$. How do we get the first row of $R$ from the given signals to start the recursion? Arrange the $2r-1$ signals $s_0, s_1, ..., s_{2r-2}$ into a Hankel matrix:

$$H = \begin{pmatrix} s_0 & s_1 & \cdots & s_{r-2} & s_{r-1} \\ s_1 & s_2 & \cdots & s_{r-1} & s_r \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_{r-2} & s_{r-1} & \cdots & s_{2r-4} & s_{2r-3} \\ s_{r-1} & s_r & \cdots & s_{2r-3} & s_{2r-2} \end{pmatrix}. \tag{2.5}$$

Using (1.1), we can verify the following decomposition:

$$H = V D_a V^T, \tag{2.6}$$

where $V$ is a Vandermonde matrix:

$$V = \begin{pmatrix} 1 & 1 & \dots & 1 \\ z_1 & z_2 & \dots & z_r \\ \vdots & \vdots & \ddots & \vdots \\ z_1^{r-1} & z_2^{r-1} & \dots & z_r^{r-1} \end{pmatrix}. \tag{2.7}$$

Comparing (2.7) with (2.1), we see that $V^T$ is a Krylov matrix. It follows from (2.4) and (2.6) that

$$H = R^T Q^T D_a Q R.$$

Since $R$ is upper triangular with a unit diagonal and $Q^T D_a Q$ is diagonal, we get a relation between the first column of $H$ and the first row of $R$:

$$H e_1 = s_0 R^T e_1.$$

Thus, the first row of $R$ equals $(1, s_1/s_0, ..., s_{r-1}/s_0)$. We summarize the procedure for computing $T$ as follows.

**Algorithm 2.1.** Given signals $s_0, s_1, ..., s_{2r-1}$, this algorithm computes the matrix $T$ in (2.3).

Initialize the previous row: $R_{\text{pre}}(1 : 2r-1) = 0$
and the current row: $R_{\text{cur}}(1 : 2r-1) = (s_0, s_1, ..., s_{2r-2})/s_0$;
$\alpha_1 = R_{\text{cur}}(2)$;
for $i = 2 : r$
$\beta_{i-1}^2 = R_{\text{cur}}(i+1) - \alpha_{i-1} R_{\text{cur}}(i) - R_{\text{pre}}(i)$;
$R_{\text{next}}(i : 2r-i) = (R_{\text{cur}}(i+1 : 2r-i+1) - \alpha_{i-1} R_{\text{cur}}(i : 2r-i) - R_{\text{prev}}(i : 2r-i))/\beta_{i-1}^2$;
$R_{\text{prev}} = R_{\text{cur}}$;
$R_{\text{cur}} = R_{\text{next}}$;
$\alpha_i = R_{\text{cur}}(i+1) - R_{\text{prev}}(i)$;
end.

Now we compute the eigenvalues of $T$. Since the original matrix $H$ is symmetric, it makes sense to symmetrize $T$ to maintain its tridiagonality, which can be achieved by a similarity transformation:

$$D_\beta T D_\beta^{-1} = \hat{T}, \tag{2.8}$$

where

$$D_\beta \equiv \operatorname{diag}\left(1,\ \beta_1,\ \beta_1\beta_2,\ ...,\ \prod_{i=1}^{r-1} \beta_i\right)$$

and

$$\hat{T} = \begin{pmatrix} \alpha_1 & \beta_1 & & & 0 \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{r-2} & \alpha_{r-1} & \beta_{r-1} \\ 0 & & & \beta_{r-1} & \alpha_r \end{pmatrix}. \tag{2.9}$$

Thus, the exponential decomposition has been reduced to an eigenproblem of a complex symmetric tridiagonal matrix.

## 3. COMPLEX-ORTHOGONAL TRANSFORMATIONS

Let us study the use of complex-orthogonal transformations for the complex symmetric eigenproblem. A matrix $Q \in C^{n\times n}$ is complex-orthogonal if

$$Q^T Q = QQ^T = I.$$

Complex-orthogonal transformations preserve a complex metric $x^T x$ of any vector $x \in C^n$; specifically, if $Q$ is complex-orthogonal and $y = Qx$, then $y^T y = x^T x$. The basic operation in solving eigenproblems is the introduction of zeros. In the following, we present two kinds of elementary complex-orthogonal transformations that serve this purpose.

First, the general form of a 2-by-2 complex-orthogonal matrix $G_2 \in C^{2\times 2}$ is

$$G_2 = \begin{pmatrix} c & s \\ \mp s & \pm c \end{pmatrix},$$

where $c^2 + s^2 = 1$. In this paper, we elect to consider the nonsymmetric version; one example of $G_2$ is

$$G_2 = \begin{pmatrix} \cosh\theta & i\sinh\theta \\ -i\sinh\theta & \cosh\theta \end{pmatrix}.$$

**Algorithm 3.1.** Given $x \in C^2$ ($x^T x \neq 0$), this method computes $G_2$ so that $G_2 x = \xi e_1$ for a scalar $\xi$.

if ($|x_1| > |x_2|$)
  $t = x_2/x_1$; $c = 1/\sqrt{1+t^2}$; $s = tc$;
else
  $t = x_1/x_2$; $s = 1/\sqrt{1+t^2}$; $c = ts$; end.

Second, we present an $n \times n$ symmetric complex-orthogonal matrix $G_n \in C^{n\times n}$ of the form:

$$G_n = I - \sigma^{-1} u u^T.$$

Given $x \in C^n$ such that $x^T x \neq 0$, Algorithm 3.2 computes $G_n$ such that

$$G_n x = -\alpha e_1.$$

It is easily verified that $\sigma = u^T u/2$. As we can see, these two types of transformations $G_2$ and $G_n$ are similar to the Givens and Householder transformations, respectively.

**Algorithm 3.2.** Given $x \in C^n$ ($x^T x \neq 0$), this algorithm computes $G_n$ so that $G_n x = -\alpha e_1$.

1. $m = \max(|x_1|, ..., |x_n|)$;
2. $v = x/m$;
3. $\hat\alpha = \sqrt{v^T v}$;
4. $u = v + \hat\alpha e_1$;
5. $\sigma = \hat\alpha u_1$;
6. $\alpha = m\hat\alpha$.

Now we consider the eigenproblem of the tridiagonal matrix $\hat{T}$ in (2.9). We use complex-orthogonal transformations to diagonalize $\hat{T}$. The process is similar to the traditional $QR$ algorithm [3]. The shift strategy can also be applied. The final decomposition is given by

$$\hat{T} = PSP^T, \tag{3.1}$$

where $P$ is complex-orthogonal and

$$S \equiv \mathrm{diag}(\sigma_1, \sigma_2, ..., \sigma_n);$$

(3.1) is similar to the symmetric eigenvalue decomposition [3], except that $P$ is complex-orthogonal and not unitary. The $\sigma_i$ are the eigenvalues of $\hat{T}$. In our case, $\hat{T}$ is tridiagonal, so the basic procedure is "bulge" chasing. Specifically, the first transformation in a QR step creates "bulges" outside the tridiagonal band. Then we restore the tridiagonal structure by chasing the "bulges" down and eventually out of the matrix.

Note that $\hat{T}$ is not Hermitian. If we use existing methods based on unitary transformations, we would destroy the tridiagonal structure and use $O(n^3)$ operations. By applying complex-orthogonal transformations, we can diagonalize $\hat{T}$ while maintaining its tridiagonality through symmetry. The computational cost is thereby reduced to $O(n^2)$ operations. Unfortunately, a complex-orthogonal matrix can be ill-conditioned. Take this example:

$$G_2 = \frac{1}{\sqrt{\epsilon^2 + 2\epsilon i}} \begin{pmatrix} \epsilon + i & 1 \\ 1 & -\epsilon - i \end{pmatrix}.$$

It has a condition number that roughly equals $2/\epsilon$, which can become arbitrarily large.

**Example 3.1.** Let

$$\hat{T} = \begin{pmatrix} 0.90 + 0.78i & 0.64 + 0.58i & & \\ 0.64 + 0.58i & -0.54 + 0.52i & -0.11 + 0.84i & \\ & -0.11 + 0.84i & 0.21 - 0.09i & 0.23 + 0.48i \\ & & 0.23 + 0.48i & -0.03 - 0.96i \end{pmatrix}.$$

We use the complex-orthogonal transformations to diagonalize $\hat{T}$ using the shifted $QR$ algorithm. After four $QR$ steps, $\hat{T}$ becomes

$$\begin{pmatrix} 0.9629 + 1.1550i & -0.1924 - 0.0328i & & \\ -0.1924 - 0.0328i & -0.2394 + 0.6090i & 0.0644 + 0.0034i & \\ & 0.0644 + 0.0034i & -0.0009 - 0.2547i & \\ & & & -0.1826 - 1.2593i \end{pmatrix}.$$

In total of seven $QR$ steps, $\hat{T}$ converges to the diagonal matrix

$$S = \mathrm{diag}(0.9911 + 1.1527i,\ -0.2685 + 0.6067i,\ -0.0001 - 0.2502i,\ -0.1826 - 1.2593i).$$

## 4. EXPONENTIAL DECOMPOSITION

A further advantage of using complex-orthogonal transformations is that we can efficiently compute the coefficients $a_i$; in the literature, the computation of $a_i$ usually involves solving a linear system [1,2,5]. Denote $a = (a_1, a_2, ..., a_r)^T$. From the definitions of the Hankel matrix $H$ in (2.5) and the Vandermonde matrix $V$ in (2.7) and the model (1.1), we get

$$Va = He_1.$$

Using the decomposition of $H$ in (2.6), we obtain

$$a = D_a V^T e_1.$$

Recalling that in the Lanczos approach, we have the $QR$ decomposition: $V^T = QR$. Thus, assuming that $Q^T D_a Q = D$ is diagonal, we deduce that

$$a = D_a Q R e_1 = D_a Q e_1 = Q^{-T} D e_1. \tag{4.1}$$

Since $D$ is diagonal, (4.1) means that the coefficient vector $a$ equals $Q^{-T} e_1$ (up to a scalar factor). Equation (2.2) shows that the columns of $Q^{-1}$ are the eigenvectors of $T$. On the other hand, it follows from (2.8) and (3.1) that we have obtained an eigenvalue decomposition of $T$:

$$T = D_\beta^{-1} P S P^T D_\beta.$$

Thus, $D_\beta^{-1} P$ is an eigenvector matrix of $T$. Assuming that the diagonal of $S$ is permuted so that $S = D_z$, then each column of $D_\beta^{-1} P$ is a scalar multiple of the corresponding column of $Q^{-1}$, i.e.,

$$(D_\beta^{-1} P) M = Q^{-1}, \tag{4.2}$$

where $M$ is a diagonal scaling matrix. Now, we determine the diagonal matrix $M$. From (2.1) and (2.4), we have $Qe_1 = e$, since $R$ is upper triangular with unit diagonal. Consequently, from (4.2) we get

$$M^{-1} P^T D_\beta e_1 = e.$$

Recalling that $D_\beta = \mathrm{diag}(1,\ \beta_1,\ ...,\ \prod_{i=1}^{r-1} \beta_i)$, the above equation implies

$$M = \mathrm{diag}(P^T e_1).$$

Finally, the coefficient vector is given by

$$a = \gamma Q^{-T} e_1 = \gamma M P^T D_\beta^{-1} e_1 = \gamma\, \mathrm{diag}(P^T e_1)\, P^T e_1 = \gamma \begin{pmatrix} P_{1,1}^2 \\ P_{1,2}^2 \\ \vdots \\ P_{1,r}^2 \end{pmatrix},$$

where $\gamma$ is a scalar, computable from $\gamma = s_0 / \sum_{i=1}^{r} P_{1,i}^2$, since $\sum_{i=1}^{r} a_i = s_0$; the exponentials $z_i$ are thus given by the diagonal elements of $S$, and the coefficients $a_i$ by $\gamma P_{1,i}^2$.

**Algorithm 4.1.** Given signals $\{s_i\}$, this algorithm computes exponentials $\{z_i\}$ and coefficients $\{a_i\}$.

1. Compute the tridiagonal $T$ in (2.3) using Algorithm 2.1;
2. Transform $T$ into the symmetric $\hat{T}$ in (2.9);
3. Diagonalize $\hat{T}$ using complex-orthogonal transformations: $\hat{T} = PSP^T$.

## 5. NUMERICAL EXPERIMENTS

Finally, we present some results of our numerical experiments with Algorithm 4.1. The algorithm is programmed in MATLAB. The construction of the input data is adopted from [1]. Consider a closed interval $[0.1,\ t]$, over which are equally spaced points $\theta_j$. The exponentials and the coefficients are respectively

$$z_j = e^{i\pi\theta_j} \quad \text{and} \quad a_j = 1,$$

for $j = 1, ..., r$. The error measurements are also adopted from [1].

Table 5.1 lists the errors in the computed exponentials for various values of $t$ and $r$. For each pair of $t$ and $r$, the error is measured as follows. Let $\hat{z}_i$ denote a computed exponential. For each $k$, we first determine $j$ such that

$$|z_j - \hat{z}_k| = \min_i(|z_i - \hat{z}_k|),$$

and then we determine $\hat{k}$ such that

$$|z_j - \hat{z}_{\hat{k}}| = \min_i(|z_j - \hat{z}_i|).$$

In words, $z_j$ is the original exponential closest to $\hat{z}_k$ and $\hat{z}_{\hat{k}}$ is the computed exponential closest to $z_j$. So, given a $\hat{z}_j$ that approximates $z_j$, we hope that there is not another $\hat{z}_{\hat{k}}$ closer to $z_j$. Thus, if $k = \hat{k}$, then $\hat{z}_k$ is accepted as an approximation of $z_j$; otherwise $\hat{z}_k$ is considered as a bad approximation and will be excluded from the exponential error calculation. The number of bad approximations is indicated in parenthesis. For example, there are eight bad approximations when $t = 0.11$ and $r = 12$. Each entry in Table 5.1 is the square root of the sum-of-squares errors in the approximated exponentials (bad approximations are excluded). Note that the number of bad approximations increases as the given exponentials become clustered.

Table 5.2 presents the average of the absolute errors in the computed coefficients $a_i$ over all $r$ coefficients. Signals are then reconstructed by using all the computed exponentials $\{z_j\}$ and the computed coefficients $\{a_j\}$. Table 5.3 displays the average of the absolute errors in the first $2r$ reconstructed signals. While Tables 5.1 and 5.2 show that the errors in the computed exponentials and coefficients become large when the exponentials are clustered in a small interval (small $t$ and large $r$), Table 5.3 shows that the reconstructed signals remain consistently accurate. We thus have obtained empirical evidence that our method is stable.

| $t$ | $r = 4$ | $r = 8$ | $r = 12$ |
|---|---|---|---|
| 0.90 | 5.0E−15(0) | 7.4E−11(0) | 1.3E−06(0) |
| 0.70 | 3.3E−14(0) | 6.1E−09(0) | 9.2E−04(0) |
| 0.50 | 1.3E−13(0) | 2.8E−06(0) | 1.4E−01(0) |
| 0.30 | 4.1E−11(0) | 5.0E−02(1) | 7.4E−02(3) |
| 0.15 | 7.4E−08(0) | 1.5E−02(2) | 1.0E−02(6) |
| 0.11 | 4.5E−04(0) | 1.7E−03(4) | 2.2E−02(8) |

**Table 5.1.** Square root of the sum-of-squares errors in the computed exponentials $\{z_i\}$.

| $t$ | $r = 4$ | $r = 8$ | $r = 12$ |
|---|---|---|---|
| 0.90 | 7.4E−15 | 1.5E−10 | 2.8E−06 |
| 0.70 | 5.6E−14 | 1.2E−08 | 2.0E−03 |
| 0.50 | 2.5E−13 | 6.0E−06 | 3.2E−01 |
| 0.30 | 1.1E−10 | 2.6E−01 | 5.4E−01 |
| 0.15 | 7.1E−07 | 5.3E−01 | 1.0E+00 |
| 0.11 | 2.2E−02 | 1.0E+00 | 1.3E+00 |

**Table 5.2.** Average absolute errors in the computed coefficients $\{a_i\}$.

| $t$ | $r = 4$ | $r = 8$ | $r = 12$ |
|---|---|---|---|
| 0.90 | 1.3E−14 | 2.8E−14 | 8.3E−10 |
| 0.70 | 1.3E−14 | 9.7E−14 | 9.1E−13 |
| 0.50 | 5.0E−15 | 5.8E−13 | 2.8E−13 |
| 0.30 | 1.2E−14 | 1.1E−13 | 1.5E−13 |
| 0.15 | 6.6E−15 | 1.1E−12 | 1.2E−12 |
| 0.11 | 6.1E−15 | 6.5E−14 | 5.3E−14 |

**Table 5.3.** Average absolute errors in the reconstructed signals $\{s_i\}$.

## ACKNOWLEDGEMENTS

The authors thank D. Boley for valuable discussions. S. Qiao was partly supported by the Natural Sciences and Engineering Research Council of Canada under grant OGP0046301.

## REFERENCES

[1] Ammar, G., W. Dayawansa, and C. Martin (1991). Exponential interpolation: Theory and numerical algorithms, *Applied Mathematics and Computation*, Vol. 41, pp. 189–232.

[2] Boley, D.L., T.J. Lee, and F.T. Luk (1992). The Lanczos algorithm and Hankel matrix factorization, *Lin. Alg. Appl.*, Vol. 172, pp. 109–133.

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

[4] Gragg, W. (1974). Matrix interpolations and applications of the continued fraction algorithm, *Rocky Mountain J. Math.*, Vol. 4, pp. 213–225.

[5] Kung, S.Y., K.S. Arun, and D.V. Bhaskar Rao (1983), State-space and singular-value decomposition-based approximation methods for the harmonic retrieval problem, *J. Opt. Soc. Am.*, Vol. 73, No. 12, pp. 1799–1811.

[6] Prony, R. (1795). Essai expérimental et analytique sur les lois de la dilatabilité et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures, *J. de l'Ecole Polytechnique*, 1:24–76.

[7] Vandevoorde, D. (1996). *A Fast Exponential Decomposition Algorithm and Its Applications to Structured Matrices*, Ph.D. Dissertation, Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY.