Calculation of 3-D nonnegative outer product expansion by the power method and its application to digital signal processing

Chiharu OKUMA*, Jun MURAKAMI* and Naoki YAMAMOTO**

Dept. of Information and Computer Sciences*,
Dept. of Information and Communication Engineering**,
Kumamoto National College of Technology,
2659-2, Suya, Koshi, Kumamoto pref., 861-1102, JAPAN E-mail:chiharu@cs.knct.ac.jp

Abstract

The power method is known as a convenient way to calculate eigenvectors of a matrix. We used this method for calculating 3-D outer product expansion previously. In this paper, we try calculating 3-D nonnegative outer product expansion by using the power method. In order to perform this calculation, we add nonnegative constraint conditions to the repetition process of the power method. Our method shows a significant reduction of computation time than the nonlinear optimization method.

key words:

3-D Outer Product Expansion, Power Method, 3-D array, Nonnegative Outer Product Expansion.

1. Introduction

In the field of image processing and digital signal processing, multi-dimensional digital filters are usually used. In order to design a multi-dimensional digital filter, multi-dimensional design specification is generally reduced to a set of 1-dimensional (1-D) specification array. Then the desired multi-dimensional filter can be obtained by designing a set of 1-D digital filters and combining them each other [1].

3-dimensional (3-D) outer product expansion [2] is usually used to decompose 3-D data arrays into products of 1-D vectors. This expansion is an extension of the singular value decomposition (SVD) of a matrix to a 3-D array. To simplify the structure of resultant 3-D digital filter, the number of terms of the expansion is desired to be reduced as much as possible. Therefore, such terms have to give least square approximation to the original 3-D array under some constraint conditions. In practice, although a large amount of computation time is required to calculate that expansion, the nonlinear optimization method is exploited ordinarily. We previously proposed the method, which uses the power method, for the purpose of calculating that 3-D outer product expansion and showed the efficiency of our method in comparison with the nonlinear optimization method [3]. The power method is known as a basic numerical technique to calculate eigenvalues of a matrix [4].

Since the 1-D vector obtained by 3-D outer product expansion represents magnitude response in the practical digital filter design problem, every elements of the vector should be physically nonnegative. We call this expansion 3-D nonnegative outer product expansion. Though the nonlinear optimization method can be used to calculate this expansion likewise as the case of ordinary 3-D outer product expansion, the computation time is the weakness of this method similarly as above.

In this paper, we try calculating 3-D nonnegative outer product expansion by using the power method. In order to perform this calculation, we add nonnegative constraint conditions to the repetition process of the power method.

2. Definition of 3-D Outer Product Expansion

A $L \times M \times N$ 3-D array $\boldsymbol{A}_3$ can be decomposed by the 3-D outer product expansion as

$$\boldsymbol{A}_3 = \sum_{i=1}^{r} \sigma_i \left(\boldsymbol{u}_i \otimes \boldsymbol{v}_i \otimes \boldsymbol{w}_i \right), \qquad (1)$$

$$\sigma_1 \ge \sigma_2 \ge \cdots \ge \sigma_r ,$$

where the expansion vectors $\boldsymbol{u}_i$, $\boldsymbol{v}_i$, $\boldsymbol{w}_i$ correspond to the singular vectors of the SVD of a matrix, the expansion coefficients σ_i and the number of expansion terms r correspond to the singular values and the rank of a matrix similarly, and $\otimes$ denotes the outer product operation. The expansion vectors are normalized as

$$\|\boldsymbol{u}_i\| = \sqrt{\sum_{j=1}^{L} \boldsymbol{u}_i(j)^2} = 1,$$

$$\|\boldsymbol{v}_i\| = \sqrt{\sum_{j=1}^{M} \boldsymbol{v}_i(j)^2} = 1, \qquad (2)$$

$$\|\boldsymbol{w}_i\| = \sqrt{\sum_{j=1}^{N} \boldsymbol{w}_i(j)^2} = 1,$$

where $\boldsymbol{u}_i(j)$, $\boldsymbol{v}_i(j)$, $\boldsymbol{w}_i(j)$ show the j-th element of the vector $\boldsymbol{u}_i$, $\boldsymbol{v}_i$, $\boldsymbol{w}_i$ respectively.

3. Calculation Algorithm for 3-D Outer Product Expansion by the Power Method

The algorithm for calculating 3-D outer product expansion by the power method is described as follows [3].

Step 1. Choose the initial vectors $\boldsymbol{u}_n{}^{(p)}$, $\boldsymbol{v}_n{}^{(p)}$, $\boldsymbol{w}_n{}^{(p)}$ arbitrarily, where these vectors must be normalized, and the subscript p and n are set to zero and one respectively at the beginning of this repetitious procedure.

Step 2. The residual 3-D array $\boldsymbol{B}_3$ is obtained by subtracting sum of products of the expansion vectors $\boldsymbol{u}_i$, $\boldsymbol{v}_i$, $\boldsymbol{w}_i$, which has been calculated by this time, from original 3-D array $\boldsymbol{A}_3$ as follows:

$$\boldsymbol{B}_3 = \boldsymbol{A}_3 - \sum_{i=1}^{n-1} \sigma_i (\boldsymbol{u}_i \otimes \boldsymbol{v}_i \otimes \boldsymbol{w}_i). \tag{3}$$

Step 3. Calculate the $L \times M$ matrix $\boldsymbol{F}$ by multiplying $\boldsymbol{B}_3$ by vector $\boldsymbol{w}_n{}^{(p)}$ as

$$\boldsymbol{F} = \boldsymbol{B}_3 \cdot \boldsymbol{w}_n{}^{(p)}. \tag{4}$$

The (i, j) element of the matrix $\boldsymbol{F}$ can be represented as

$$\boldsymbol{F}(i, j) = \sum_k \boldsymbol{B}_3(i, j, k) \boldsymbol{w}_n{}^{(p)}(k). \tag{5}$$

Next, apply the power method to the matrix $\boldsymbol{F}$ as follows:

$$\begin{aligned} \boldsymbol{u}_n{}^{(p+1)} &= \boldsymbol{F}\boldsymbol{v}_n{}^{(p)}, \\ \boldsymbol{v}_n{}^{(p+1)} &= \boldsymbol{F}^T \boldsymbol{u}_n{}^{(p+1)}. \end{aligned} \tag{6}$$

Like wise the $M \times N$ matrix $\boldsymbol{G}$ and the $N \times L$ matrix $\boldsymbol{H}$ are obtained by

$$\begin{aligned} \boldsymbol{G} &= \boldsymbol{B}_3 \cdot \boldsymbol{v}_n{}^{(p+1)}, \\ \boldsymbol{w}_n{}^{(p+1)} &= \boldsymbol{G}\boldsymbol{u}_n{}^{(p+1)}, \\ \boldsymbol{u}_n{}^{(p+1)} &= \boldsymbol{G}^{\boldsymbol{T}} \boldsymbol{w}_n{}^{(p+1)}, \\ \boldsymbol{H} &= \boldsymbol{B}_3 \cdot \boldsymbol{u}_n{}^{(p+1)}, \\ \boldsymbol{v}_n{}^{(p+1)} &= \boldsymbol{H}\boldsymbol{u}_n{}^{(p+1)}, \\ \boldsymbol{w}_n{}^{(p+1)} &= \boldsymbol{H}^{\boldsymbol{T}} \boldsymbol{w}_n{}^{(p+1)}, \end{aligned} \tag{7}$$

where the obtained vectors $\boldsymbol{u}_n{}^{(p+1)}$, $\boldsymbol{v}_n{}^{(p+1)}$, $\boldsymbol{w}_n{}^{(p+1)}$ must be normalized.

Repeat Step 3 until the following are satisfied for sufficiently small value ε:

$$\begin{cases} \left\| \boldsymbol{u}_n{}^{(p+1)} - \boldsymbol{u}_n^{(p)} \right\| < \varepsilon, \\ \left\| \boldsymbol{v}_n{}^{(p+1)} - \boldsymbol{v}_n^{(p)} \right\| < \varepsilon, \\ \left\| \boldsymbol{w}_n{}^{(p+1)} - \boldsymbol{w}_n^{(p)} \right\| < \varepsilon. \end{cases} \tag{8}$$

Step 4. The n-th expansion vectors $\boldsymbol{u}_n{}^{(p+1)}$, $\boldsymbol{v}_n{}^{(p+1)}$, $\boldsymbol{w}_n{}^{(p+1)}$ are obtained from Step 3. Here, rename these vectors as $\boldsymbol{u}_n$, $\boldsymbol{v}_n$, $\boldsymbol{w}_n$.

The n-th coefficient σ_n is obtained from inner product operation as as

$$\sigma_n = \boldsymbol{B}_3 (\boldsymbol{u}_n \otimes \boldsymbol{v}_n \otimes \boldsymbol{w}_n). \tag{10}$$

Step 5. After increase n and set p to zero, repeat this procedure from Step 1.

4. Calculation of 3-D Nonnegative Outer Product Expansion by the Power Method

The method which we described in Section 3 can be applied to calculation of a 3-D nonnegative outer product expansion by adding nonnegative constraint conditions to the repetition process of the power method. Actually, the following steps are inserted into Step 3.

Step A1. The repetition vector $\boldsymbol{u}_n{}^{(p+1)}$ in equation (6) is divided into the vector of $\boldsymbol{u}_n{}^{(p+1)+}$ and $\boldsymbol{u}_n{}^{(p+1)-}$, where the former is composed of positive number or zero and the latter is composed of negative number or zero as

$$\begin{aligned} \boldsymbol{u}_n{}^{(p+1)} &= \boldsymbol{u}_n{}^{(p+1)+} - \boldsymbol{u}_n{}^{(p+1)-}, \\ \boldsymbol{u}_n{}^{(p+1)+} &= \max(\boldsymbol{u}_n{}^{(p+1)}, 0), \\ \boldsymbol{u}_n{}^{(p+1)-} &= -\min(\boldsymbol{u}_n{}^{(p+1)}, 0). \end{aligned} \tag{11}$$

Step A2. Calculate the norm of $\boldsymbol{u}_n{}^{(p+1)+}$ and $\boldsymbol{u}_n{}^{(p+1)-}$. Choose the nonnegative vector $\boldsymbol{u}_n{}^{(p+1)}$ from these vectors by following way.

$$\boldsymbol{u}_n{}^{(p+1)} = \begin{cases} \boldsymbol{u}_n{}^{(p+1)+} & \left(\left\| \boldsymbol{u}_n{}^{(p+1)+} \right\| \ge \left\| \boldsymbol{u}_n{}^{(p+1)-} \right\| \right) \\ -\boldsymbol{u}_n{}^{(p+1)-} & \left(\left\| \boldsymbol{u}_n{}^{(p+1)+} \right\| < \left\| \boldsymbol{u}_n{}^{(p+1)-} \right\| \right). \end{cases} \tag{12}$$

The vector $\boldsymbol{v}_n{}^{(p+1)}$ and $\boldsymbol{w}_n{}^{(p+1)}$ are also chosen as follows:

$$\boldsymbol{v}_n{}^{(p+1)} = \begin{cases} \boldsymbol{v}_n{}^{(p+1)+} & \left(\left\| \boldsymbol{v}_n{}^{(p+1)+} \right\| \ge \left\| \boldsymbol{v}_n{}^{(p+1)-} \right\| \right) \\ -\boldsymbol{v}_n{}^{(p+1)-} & \left(\left\| \boldsymbol{v}_n{}^{(p+1)+} \right\| < \left\| \boldsymbol{v}_n{}^{(p+1)-} \right\| \right), \end{cases} \tag{13}$$

$$\boldsymbol{w}_n{}^{(p+1)} = \begin{cases} \boldsymbol{w}_n{}^{(p+1)+} & \left(\left\| \boldsymbol{w}_n{}^{(p+1)+} \right\| \ge \left\| \boldsymbol{w}_n{}^{(p+1)-} \right\| \right) \\ -\boldsymbol{w}_n{}^{(p+1)-} & \left(\left\| \boldsymbol{w}_n{}^{(p+1)+} \right\| < \left\| \boldsymbol{w}_n{}^{(p+1)-} \right\| \right). \end{cases} \tag{14}$$

5. Calculation of 3-D Orthogonal Outer Product Expansion

Since the resultant expansion terms of 3-D outer product expansion do not satisfy orthogonality, the 3-D orthogonal outer product expansion [3] is defined by

$$\mathbf{A}_3 = \sum_{i,j,k} \sigma_{ijk} (\mathbf{u}_i \otimes \mathbf{v}_j \otimes \mathbf{w}_k), \tag{15}$$

where σ_{ijk} are the expansion coefficients. This expansion can be calculated by introducing the Gram-Schmidt orthogonalization process [5] into the Step 3 of the algorithm described in Section 3 as following procedure.

Step B1. Along with the Gram-Schmidt process, calculate the vectors $\boldsymbol{u}'_n{}^{(p+1)}$, $\boldsymbol{v}'_n{}^{(p+1)}$, $\boldsymbol{w}'_n{}^{(p+1)}$ by subtracting the previously obtained quantities from vectors $\boldsymbol{u}_n{}^{(p+1)}$, $\boldsymbol{v}_n{}^{(p+1)}$, $\boldsymbol{w}_n{}^{(p+1)}$ respectively as

$$\boldsymbol{u}'_n{}^{(p+1)} = \boldsymbol{u}_n{}^{(p+1)} - \left(\boldsymbol{u}_1{}^T \boldsymbol{u}_n{}^{(p+1)}\right)\boldsymbol{u}_1 - \left(\boldsymbol{u}_2{}^T \boldsymbol{u}_n{}^{(p+1)}\right)\boldsymbol{u}_2 - \cdots - \left(\boldsymbol{u}_{n-1}{}^T \boldsymbol{u}_n{}^{(p+1)}\right)\boldsymbol{u}_{n-1}, \tag{16}$$

$$\boldsymbol{v}'_n{}^{(p+1)} = \boldsymbol{v}_n{}^{(p+1)} - \left(\boldsymbol{v}_1{}^T \boldsymbol{v}_n{}^{(p+1)}\right)\boldsymbol{v}_1 - \left(\boldsymbol{v}_2{}^T \boldsymbol{v}_n{}^{(p+1)}\right)\boldsymbol{v}_2 - \cdots - \left(\boldsymbol{v}_{n-1}{}^T \boldsymbol{v}_n{}^{(p+1)}\right)\boldsymbol{v}_{n-1}, \tag{17}$$

$$\boldsymbol{w}'_n{}^{(p+1)} = \boldsymbol{w}_n{}^{(p+1)} - \left(\boldsymbol{w}_1{}^T \boldsymbol{w}_n{}^{(p+1)}\right)\boldsymbol{w}_1 - \left(\boldsymbol{w}_2{}^T \boldsymbol{w}_n{}^{(p+1)}\right)\boldsymbol{w}_2 - \cdots - \left(\boldsymbol{w}_{n-1}{}^T \boldsymbol{w}_n{}^{(p+1)}\right)\boldsymbol{w}_{n-1}. \tag{18}$$

Normalize the vectors in above equations to obtain $\boldsymbol{u}_n{}^{(p+1)}$, $\boldsymbol{v}_n{}^{(p+1)}$, $\boldsymbol{w}_n{}^{(p+1)}$.

Step B2. By the procedure in Section 3 and the Step B1 in this section, vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m$ of the equation (15) are obtained in order, where $m = \min(L,M,N)$. In case that $L>m$, the remaining $L-m$ vectors terms can be calculated by using Gram-Schmidt orthogonalization process as

$$\boldsymbol{u}_n{}' = \boldsymbol{u}_n - (\boldsymbol{u}_1{}^T \boldsymbol{a}_n)\boldsymbol{u}_1 - (\boldsymbol{u}_2{}^T \boldsymbol{u}_n)\boldsymbol{u}_2 \cdots - (\boldsymbol{u}_{n-1}{}^T \boldsymbol{u}_n)\boldsymbol{u}_{n-1}, \tag{19}$$

$$n = m+1, \cdots, L$$

where $\boldsymbol{u}_n$ are the initial vectors and the vectors $\boldsymbol{u}_n{}'$ are to renamed as $\boldsymbol{u}_n$ after they are normalized. Likewise vectors $\boldsymbol{v}_{m+1}, \cdots, \boldsymbol{v}_M$ and $\boldsymbol{w}_{m+1}, \cdots, \boldsymbol{w}_N$ are calculated.

Step B3. For every combination of p, q and r, calculate the expansion coefficients σ_{pqr} as

$$\sigma_{pqr} = \boldsymbol{A}_3\left(\boldsymbol{u}_p \otimes \boldsymbol{v}_q \otimes \boldsymbol{w}_r\right), \tag{20}$$

$$(\, p = 1,2,\cdots L\,,\ \ q = 1,2,\cdots M\,,\ \ r = 1,2,\cdots N\,).$$

To improve in calculation time of these steps, a part of the Step B1 is modified. The modification is described below.

After the calculation of the expansion vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_{m-1}$, the remaining vector $\boldsymbol{u}_m$ can be calculated by

$$\boldsymbol{u}_m{}' = \boldsymbol{u}_m - (\boldsymbol{u}_1{}^T \boldsymbol{a}_n)\boldsymbol{u}_1 - (\boldsymbol{u}_2{}^T \boldsymbol{u}_n)\boldsymbol{u}_2 \cdots - (\boldsymbol{u}_{m-1}{}^T \boldsymbol{u}_m)\boldsymbol{u}_{m-1}, \tag{21}$$

where $\boldsymbol{u}_m$ is the initial vector. The vector $\boldsymbol{u}_m{}'$ is normalized immediately, then the vector renamed as $\boldsymbol{u}_m$. This slight modification leads to an improvement in calculation time.

6. Experimental Results

The following magnitude specification $\boldsymbol{h}_d(x_i, y_j, z_k)$ of a 3-D digital filter design problem [2] is used to consider the validity of calculation algorithm described above.

$$\boldsymbol{h}_d(x_i, y_j, z_k) = \begin{cases} 1, & (0 \le r \le 0.4) \\ \dfrac{(0.6-r)}{0.2}, & (0.4 \le r \le 0.6) \\ 0, & (r \ge 0.6), \end{cases} \tag{22}$$

where

$$r = \frac{1}{\pi}\sqrt{x_i^2 + y_j^2 + z_k^2},$$

$$\left.\begin{aligned} x_i &= \frac{i\pi}{L'-1}, (0 \le i \le L'-1), \\ y_j &= \frac{j\pi}{M'-1}, (0 \le j \le M'-1), \\ z_k &= \frac{k\pi}{N'-1}, (0 \le k \le N'-1). \end{aligned}\right\} \tag{23}$$

The 3-D array $\boldsymbol{A}_3$ is constituted by

$$\boldsymbol{A}_3(i,j,k) = \boldsymbol{h}_d(x_i, y_j, z_k). \tag{24}$$

Since the magnitude specification $\boldsymbol{h}_d(x_i, y_j, z_k)$ is zero when $r \ge 0.6$, the size of the 3-D array is reduced to $L \times M \times N$, where $L = L' \times 0.6$, $M = M' \times 0.6$, $N = N' \times 0.6$.

[3-D Outer Product Expansion]

Table 1 shows that the calculated expansion coefficients by the power method give good approximation to those by the nonlinear optimization method.

Table 1. Resultant coefficients of 3-D outer product expansion by the nonlinear optimization method and the power method.

($L'=M'=N'=20$)

σ_i	Nonlinear Optimization Method	Power Method	Relative Error[%]
1	2.275862E+01	2.275862E+01	0
2	4.283573E+00	4.283573E+00	0
3	3.025678E+00	3.025678E+00	0
4	1.400982E+00	1.400982E+00	0
5	1.129514E+00	1.129514E+00	0
6	6.526013E-01	6.526013E-01	0
7	3.698252E-01	3.698252E-01	0
8	3.454422E-01	3.454422E-01	0
9	3.403220E-01	3.403220E-01	0
10	2.840046E-01	2.840046E-01	0
11	2.598434E-01	2.598434E-01	0
12	2.108770E-01	2.108770E-01	0
13	2.008905E-01	2.008905E-01	0
14	1.767691E-01	1.767691E-01	0
15	1.535205E-01	1.535205E-01	0
16	1.494289E-01	1.494289E-01	0
17	1.064934E-01	1.064934E-01	0
18	1.006265E-01	1.006265E-01	0
19	1.002772E-01	1.002772E-01	0
20	9.568394E-02	9.568395E-02	-6.27E-06

[3-D Nonnegative Outer Product Expansion]

Figure 1 illustrates the convergence property of the power method described in Section 4. The relative error of the method, which calculates 3-D nonnegative outer product expansion, is about 10% at n=10, while the error less than 10% at n=3 in case of 3-D outer product expansion.

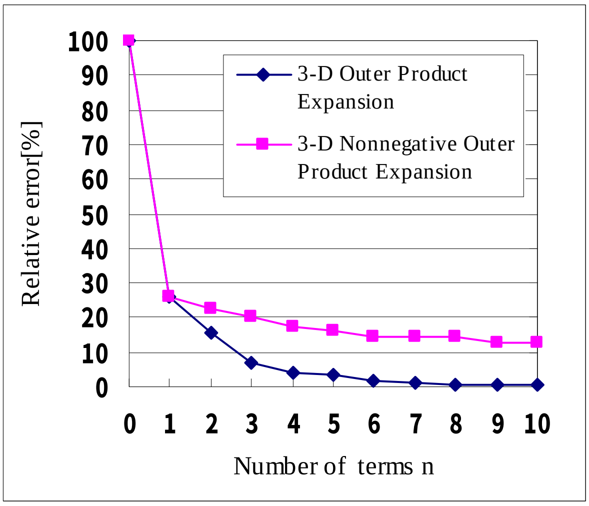

Figure 1. Convergence property of the power method.

[3-D Orthogonal Outer Product Expansion]

Figure 2 shows the calculation time of modified method in Section 5 compared with the usual method. In this experiments, the 3-D array is constructed by random integer in the range of [1,1000]. From the figure, the calculation time can be reduced slightly by using the proposed method.

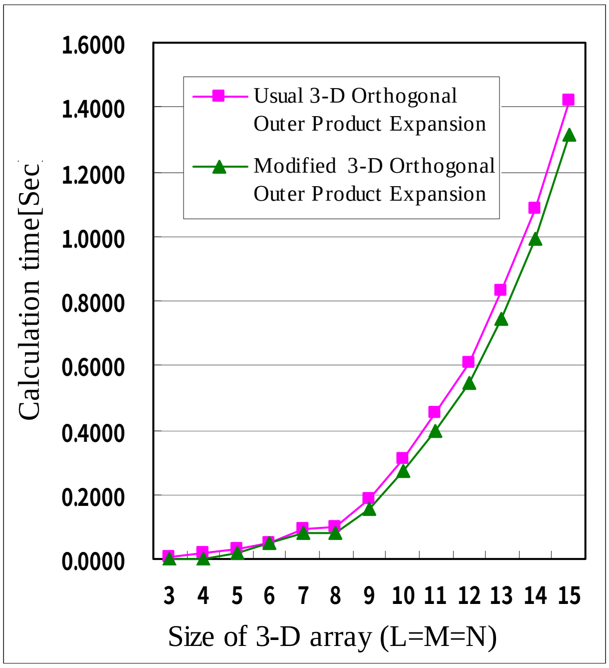

Figure 2 Calculation time of 3-D orthogonal outer product expansion.

7. Conclusions

In this paper, we showed the calculation results of the 3-D outer product expansion, 3-D nonnegative outer product expansion, and 3-D orthogonal outer product expansion in the following development environment.

System: FreeBSD 3.4-RELEASE
Development language:
C (GNU C compiler v2.7)
Accuracy:
floating point of double precision type

The results lead to the following conclusions:

(1) Using the previously proposed method to calculate the 3-D outer product expansion and the 3-D nonnegative outer product expansion, we obtained the expansion coefficients and vectors precisely.

(2) By modifying the usual method for calculation of the 3-D orthogonal outer product expansion, the calculation time could be reduced slightly in comparison with the previously proposed method.

References

[1]Deng, T. B. and Kawamata, M., ”Design of Two-Dimensional Recursive Digital Filters Based on the Iterative Singular Value Decomposition”, Trans. IEICE, Vol.E73, No.6, pp.882-892,1990.

[2]Ohki, M. and Kawamata, M., ”Design of Three-Dimensional Digital Filters Based on the Outer Product Expansion”, IEEE Trans. Circuits Syst., Vol.CAS-37, No.9, pp.1164-1167, 1990.

[3]Murakami, J., Yamamoto, N. and Tadokoro, Y., ”High-Speed Computation of 3D Tensor Product Expansion by the Power Method”, Electronics and Communications in Japan, Part 3, Vol.85, pp.63-72,2002.

[4]Wilkinson, J. H., “The Algebraic Eigenvalue Problem”, Oxford University Press, 1965.

[5]Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T., “Numerical Recipes in C”, Cambridge University Press, 1988.