Rational and Radical Fixed Point Functions for the Eigenvalue Problem and Polynomials

Jawad A. K. Hasan(2) and Mohammed A. Hasan(1)
(1)Department of Electrical & Computer Engineering, University of Minnesota Duluth
(2)Department of Electrical Engineering, University of Baghdad, Baghdad, IRAQ
E.mail:mhasan@d.umn.edu

Abstract

The derivation and implementation of many algorithms in signal/image processing and control involve some form of polynomial root-finding and/or matrix eigendecomposition. In this paper, higher order fixed point functions in rational and/or radical forms are developed. This set of iterations can be considered as extensions of known methods such as Newton's, Lagurre's and Halley's methods and can be applied to compute all zeros of a polynomial as well as all eigenvalues of a complex matrix. One of the main features of the proposed agorithms is that they could have any predetermined rate of convergence regardless of the multiplicity of the zeros or eigenvalues. Additionally, eigenvalues and eigenvectors are computed using fast matrix inverse free algorithms which are based on the QR factorization.

1. Introduction

Finding eigenvalues of systems or the zeros of a given function is a classical problem in applied sciences. Many practical problems in physics and engineering lead to eigenvalue problems. For example, high-resolution spectral estimators in signal processing and control make use of the subspace information obtained from the eigendecomposition of the covariance matrix. Most spectral estimators in signal/image processing utilize polynomial root-finding. Localization of the zeros or poles of polynomials or eigenvalues of matrices in a specific region in the complex plane such as half-planes or the exterior or interior of a circle is an important problem in studying stability of certain systems in control theory and signal processing.

There is a multitude of algorithms for factoring a polynomial into linear or quadratic factors. These methods range from locally convergent such as Newton, Lin-Barstow methods [1]-[2], and globally convergent such as the methods of Graeffe, Bernoulli and the *qd* algorithm [3]-[6]. A fast transversal filter for solving a polynomial equation is given in [7]. A comprehensive treatment of Newton method and its convergence can be found in [8]. For a survey of some of these methods the reader is referred to [9]-[13] and the references therein.

The main objective here is to develop numerical methods for computing the zeros of smooth functions or eigenvalues of matrices such that these methods are parameterized by the rate of convergence regardless of the multiplicity of zeros or eigenvalues.

2. Rational Fixed Point Iterations

In this paper, we generalize Newton, Laguerre and Halley methods [1, 2, 8] to generate higher order fixed point iterations in rational and radical forms. Specifically, given a function f, a class of functions $G_i(f, f', \cdots, f^{(r)}, g, g', \cdots, g^{(s)})$, $i = 1, 2$ will be developed so that

$$\Phi(z) = z - \frac{G_1(f, f', \cdots, f^{(r)}, g, g', \cdots, g^{(s)})}{G_2(f, f', \cdots, f^{(r)}, g, g', \cdots, g^{(s)})}, \tag{1}$$

defines an Nth order fixed point function. Here N is dependent on r and s. The functions G_1 and G_2 can be chosen so that (1) yields second order convergent iterations as shown in the following proposition.

Proposition 1. *Let f be a polynomial with simple zeros and let G be an analytic function of two variables in an open set containing the zeros of f. The fixed point function*

$$\Phi(z) = z - \frac{f}{G(f, f')} \tag{2}$$

defines a second order iteration if and only if $G(f, f')(z_i) = f'(z_i)$ for each z_i such that $f(z_i) = 0$.

Proof. The result can be obtained by verifying that under the stated conditions $\Phi(z_i) = z_i$ and $\Phi'(z_i) = 0$.

Q.E.D.

From this result, it follows that $G = f' + fF(f, f')$ for some F. When $F = 0$ this reduces to the standard Newton method. The advantage of this observation is that (2) remains at least second order iteration for any choice of F. The free parameter F can be chosen to generate fixed point iterations of any desired order. The fixed point iteration $\Phi(z)$ in Proposition 1 and many other higher order fixed point functions in rational forms can be developed based on the following result.

Theorem 2. *Let f and g be two polynomials, then the iteration $\Phi(z) = z - \frac{f(z)}{g(z)}$ is an rth order fixed point function of the zeros of f iff $g^{(i)}(z_j) = \frac{f^{(i+1)}(z_j)}{i+1}$ for $j = 1, \cdots, m$ and $i = 0, \cdots, r-1$. Here $\{z_i\}_{i=1}^m$ is the set of zeros of f.*

Proof. This result can be obtained by showing $\Phi(z_i) = z_i$, and $\Phi^{(l)}(z_i) = 0$, for $i = 1, \cdots, m$ and $l = 1, \cdots, r-1$.

Q.E.D.

Hence if $r = 1$ we obtain $g(z_i) = f'(z_i)$ and consequently $g(z) = f'(z) + h(z)f(z)$ for some function h. One can choose $h(z)$ so that Φ is third order. It turns out that $h(z_i) = \frac{-f''(z_i)}{2f'(z_i)}$ for $j = 1, \cdots, m$. Therefore, $h(z)$ can be written as $h(z) = \frac{-f''(z)+h_1(z)f(z)}{2f'(z)+h_2(z)f(z)}$, or $h(z) = \frac{-f''(z)}{2f'(z)} + h_3(z)f(z)$ for some arbitrary functions h_1, h_2 and h_3.

An alternative approach for deriving higher order rational fixed point functions can be described in the following result.

Theorem 3. *Let $\{z_i\}_{i=1}^m$ and $\{a_i\}_{i=1}^m$ be sets of non-zero complex numbers and $\{z_i\}_{i=1}^m$ is distinct, then for any $r \geq 1$*

$$z_i - \frac{c_1 + \sum_{j=1}^m \frac{a_j}{z_j^{r-1}}}{c_2 + \sum_{j=1}^m \frac{a_j}{z_j^r}} = O(z_i)^{r+1}. \qquad (3)$$

Proof. The proof can be obtained by simple manipulation of the right hand side of (3), i.e.,

$$z_i - \frac{c_1 + \sum_{j=1}^m \frac{a_j}{z_j^{r-1}}}{c_2 + \sum_{j=1}^m \frac{a_j}{z_j^r}}$$
$$= -\frac{z_i^{r+1}\{c_1 - c_2 z_i^{r+1} + \sum_{j\neq i} \frac{a_j}{a_i}(z_i - z_j)\frac{1}{z_j^r}\}}{c_2 z_i^r + a_i + \sum_{j\neq i} \frac{a_j}{a_i}(\frac{z_i}{z_j})^r} = O(z_i)^{r+1}. \qquad (4)$$

Q.E.D.

Corollary 4. *Let $\{z_i\}_{i=1}^m$ be a set of distinct non-zero complex numbers, then for any $r \geq 1$*

$$z_i - \frac{\sum_{j=1}^m \frac{1}{z_j^{r-1}}}{\sum_{j=1}^m \frac{1}{z_j^r}} = \frac{z_i^r \sum_{j\neq i}(z_i - z_j)\frac{1}{z_j^r}}{1 + \sum_{j\neq i}(\frac{z_i}{z_j})^r}. \qquad (5)$$

Let f and g be two polynomials so that the zeros of f are simple, then $\frac{g(z)}{f(z}$ can be written in partial fractions form as $\frac{g(z)}{f(z} = f_1(z) + \sum_{j=1}^m \frac{a_j}{z-z_j}$, where $\{z_i\}_{i=1}^m$ is the set of zeros of f. Hence

$$(\frac{g(z)}{f(z)})^{(r)} = f_1^{(r)}(z) + \sum_{j=1}^m (-1)^r r! \frac{a_j}{(z-z_j)^{r+1}}. \qquad (6)$$

Combining (6) with Theorem 3, we obtain the following result.

Theorem 5. *Let $\{z_i\}_{i=1}^m$ and $\{a_i\}_{i=1}^m$ be sets of non-zero complex numbers and $\{z_i\}_{i=1}^m$ be distinct. For each $r \geq 2$, the iteration*

$$w_{n+1} = w_n - \frac{f_1(w_n) + \sum_{j=1}^m \frac{a_j}{(w_n - z_j)^{r-1}}}{f_2(w_n) + \sum_{j=1}^m \frac{a_j}{(w_n - z_j)^r}} \qquad (7)$$

is $(r+1)$th order for any choice of f_1 and f_2. Hence the iteration

$$\Phi(z) = z + \frac{\frac{1}{(r-1)!}(\frac{g}{f})^{(r-1)}}{\frac{1}{r!}(\frac{g}{f})^{(r)}} \qquad (8)$$

is $(r+1)$th order for any choice of g. Moreover if $g = f'$, then we obtain the following fixed point function

$$\Phi(z) = z - \frac{\frac{1}{(r-1)!}(\frac{f'}{f})^{(r-1)}}{\frac{1}{r!}(\frac{f'}{f})^{(r)}} \qquad (9)$$

which is $(r+1)$th order.

Note that the iteration of the last result has the stated order of convergence regardless of the multiplicity of zeros provided $r \geq 2$.

Example 1. For $r = 2$, (8) reduces to

$$\Phi(z) = z - \frac{f}{f' - \frac{f'' - \frac{g''}{g}f}{2f' - 2\frac{g'}{g}f}f}. \qquad (10)$$

This follows from the observation $(\frac{g}{f})' = \frac{gf' + f'g}{f^2}$ and

$$(\frac{g}{f})'' = \frac{2f'^2 g - 2ff'g' - ff''g + f^2 g''}{f^3}.$$

Example 2. Consider the equation $f(z) = z^2 - A = 0$. If $g(z) = z^2 + A$ and $r = 2$, then we obtain the following third order fixed point function

$$\Phi(z) = \frac{3zA^2 + z^3 A}{A^2 + 3z^2 A}. \qquad (11)$$

Remark: Let f be a differentiable function, then the fixed point function of Newton's method for solving $f(z) = 0$ is $\Phi(z) = z - \frac{f(z)}{f'(z)}$. This iteration can be considered as special case of higher order fixed point functions of the form $\Phi(z) = z - \frac{f(z)}{g(z)}$. If $g^{(i)}(z_j) = \frac{f^{(i+1)}}{i+1}(z_j)$ for $j = 1, \cdots, m$ and $i = 0, \cdots, r-1$, then $\Phi(z)$ is rth order. To choose $g(z)$ so that Φ is third order iteration of the form

$$\Phi(z) = z - \frac{f}{f' - \frac{f'' - \frac{u''}{u}f}{2f' - 2\frac{u'}{u}f}f}, \qquad (12)$$

for some function u. Thus to derive a third order rational fixed point function of the form $\Phi(z) = z - \frac{f(z)}{g(z)}$, we must solve the first order differential equation

$$\frac{g'}{g} = \frac{f'' - \frac{u''}{u}f}{2f' - 2\frac{u'}{u}f}.$$

If follows that $g(z) = Au(z)\sqrt{(\frac{f(z)}{u(z)})'}$. The well known third order Halley's method can obtained by setting $A = 1$ and $u(z) = 1$. This means that if we apply Newton's method to $\frac{f}{Au(z)\sqrt{(\frac{f(z)}{u(z)})'}}$, we obtain a third order method regardless of the choice of A and $u(z)$. This freedom in choosing $u(z)$ can be utilized to derive 4th order iteration, however that entails solving a more complicated differential equation.

3. Mixed Radical and Rational Fixed Point Iteration

In this section, the objective is to derive iterations of radical form. Specifically, we are looking for rth order fixed point functions of the form $\Phi(z) = z - \sqrt[l]{G(f, f', \cdots, f^{(r-1)}, g, g', \cdots, g^{(s)})}$ for some integers r, s, l. This new class of methods is actually based on the following fixed point function of f. Iterations of mixed types can be obtained as follows.

Theorem 6 [2]. *Let f and g be two polynomials such that g is not a scalar multiple of f, then for any positive integers r and l such that $r > l$ the iteration*

$$\Phi(z) = z - \sqrt[r-l]{\frac{r!(\frac{g}{f})^{(l)}(-1)^l}{l!(\frac{g}{f})^{(r)}(-1)^r}}, \qquad (13)$$

is of order $r + l - 1$. Here $(\frac{g}{f})^{(l)}$ denotes the l derivative of $\frac{g}{f}$.

If this result is applied to solve the equation $\sin(z) = 0$, then we obtain the following iteration

$$\Phi(z) = z - \frac{\sin(z)}{\sqrt[3]{\cos(z)}}$$

which is fifth order.

One can verify that the iterations

$$\Phi(z) = z - \frac{f(z)}{f'(z) - \ln(1 + \frac{f''(z)f(z)}{2f'(z)})},$$

$$\Phi(z) = z - \frac{f(z)}{f'(z) - \sin(\frac{f''(z)f(z)}{2f'(z)})},$$

and

$$\Phi(z) = z - \frac{f(z)}{f'(z) - \tan(\frac{f''(z)f(z)}{2f'(z)})}$$

are third order.

4. Applications to Matrix Eigenvalues

Several popular algorithms for the location and computation of eigenvalues and eigenvectors are presented in [1], [9]-[10], which include, among many others, the LR algorithm, the QR algorithm for matrix eigenvalues and the Golub-Reinsch algorithm for computing the singular value decomposition. In this section we propose mixed radical and rational fixed point iteration. Some of the proposed approaches will lead to the development of new eigen-solver methods of arbitrary order and which are interestingly based only on evaluating the trace of certain matrices.

The higher order iterations described in the previous sections can also be utilized to determine the eigenvalues of matrices. For example, the iteration

$$w_{n+1} = w_n - \frac{1}{trace\{(w_n I_p - A)^{-1}\}} \tag{14}$$

is quadratically convergent to an eigenvalue of A provided that all eigenvalues are simple. This can be viewed as the matrix version of the conventional Newton method. Similarly,

$$w_{n+1} = w_n - \frac{trace\{(w_n I_p - A)^{-1}\}}{trace\{(w_n I_p - A)^{-2}\}} \tag{15}$$

is at least cubically convergent regardless of the multiplicity of the eigenvalues of A.

Theorem 5 can be applied to derive higher order iterations for computing eigenvalues of matrices as in the following results.

Theorem 7. *Let A be an $p \times p$ matrix, then for each $r \geq 2$, and for almost all w_0, the iteration*

$$w_{n+1} = w_n - \frac{trace\{(w_n I_p - A)^{-r+1}\}}{trace\{(w_n I_p - A)^{-r}\}} \tag{16}$$

converges to some eigenvalue λ_i of A and is at least of order $r+1$.

Proof. Since $trace(w_n I_p - A)^{-r} = \sum_{i=1}^{m} \frac{k_i}{(w_n - \lambda_i)^r}$, where k_i is the multiplicity of λ_i, it follows that

$$\begin{aligned} w_{n+1} - \lambda_j &= w_n - \lambda_j - \frac{trace\{(w_n I_p - A)^{-r+1}\}}{trace\{(w_n I_p - A)^{-r}\}} \\ &= w_n - \lambda_j - \frac{\sum_{i=1}^{m} \frac{k_i}{(w_n - \lambda_i)^{r-1}}}{\sum_{i=1}^{m} \frac{k_i}{(w_n - \lambda_i)^{r}}} = O(w_n - \lambda_j)^r. \end{aligned} \tag{17}$$

Q.E.D.

The main concern in Iteration (16) is that it requires matrix inversion. Efficient approaches to avoid matrix inversion are given in Algorithms 1-3.

It is also possible to develop higher order iterations by using other entries information of the inverse matrix other than the trace as described in the following result.

Theorem 8. *Let $B_r(z) = (zI_p - A)^{-r} = [B_{ij}^{(r)}(z)]$, then for each $1 \leq i, j \leq p$, the function $\phi(z) = z - \frac{B_{ij}^{(r-1)}(z)}{B_{ij}^{(r)}(z)}$ defines an $(r+1)$th order fixed point function provided that $B_{ij}^{(r)}(z)$ is not identically zero, i.e., the iteration $w_{n+1} = w_n - \frac{B_{ij}^{(r-1)}(w_n)}{B_{ij}^{(r)}(w_n)}$ converges to an eigenvalue of A.*

Proof. Let $\lambda_1, \cdots, \lambda_m$ be the set of eigenvalues of A, then there exist scalars $a_{ijk} \in \mathcal{C}$ [12, 13] independent of r such that

$$B_{ij}^{(r-1)}(z) = \sum_{k=1}^{m} a_{ijk}(z - \lambda_k)^{-r}. \tag{18}$$

The conclusion follows directly from Theorem 5.

Q.E.D.

One can also apply Theorem 5 to obtain fixed point iteration of radical forms. For any $r \geq 1$, the following iteration

$$w_{n+1} = w_n - \frac{1}{\sqrt[r]{trace(w_n I_p - A)^{-r}}}, \tag{19}$$

is rth order convergent provided all eigenvalues are simple. For the general case, the iteration

$$w_{n+1} = w_n - \frac{1}{\sqrt[r-1]{\frac{trace(w_n I_p - A)^{-r}}{trace(w_n I_p - A)^{-1}}}} \tag{20}$$

is at least $(r-1)$th order convergent regardless of the multiplicities of the eigenvalues of A.

5. Matrix Inverse Free Algorithms

The main problem in the implementations of Theorem 6-8 is that they require computing of the trace of matrix inverses. Thus we present here another important set of iterations which don't require matrix inverse computation as in the following:

$$w_{n+1} = w_n - \frac{\mathbf{b}^*(w_n I_p - A)^{-r+1}\mathbf{c}}{\mathbf{b}^*(w_n I_p - A)^{-r}\mathbf{c}}. \tag{21}$$

From Theorem 8 this is an $(r+1)$th order iteration that converges to an eigenvalue of A. Here $\mathbf{b}$ and $\mathbf{c}$ are almost arbitrary nonzero $p \times 1$ vectors. In the next algorithm, we will completely alleviate the problem of matrix inverse computation using the QR factorization.

Algorithm 1

$$\begin{aligned} \begin{bmatrix} (w_n I_p - A)^r \\ \mathbf{b}^* \end{bmatrix} &= \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} R_k \\ 0 \end{bmatrix}, \text{ QR factorization} \\ w_{n+1} &= w_n - \frac{Q_{12}^*(w_n I_p - A)\mathbf{c}}{Q_{12}^*\mathbf{c}}, \end{aligned} \tag{22}$$

where w_1 is arbitrarily chosen.

Algorithm 2

$$\begin{aligned} \begin{bmatrix} (w_n I_p - A)^r \\ U_{n-1}^* \end{bmatrix} &= \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} R_k \\ 0 \end{bmatrix}, \text{ QR factorization} \\ U_n &= Q_{12} \\ w_{n+1} &= w_n - \frac{U_n^*(w_n I - A)U_n}{U_n^* U_n}, \end{aligned} \tag{23}$$

where w_1 and U_0 are arbitrarily chosen. It is not immediately clear what convergence properties this algorithm has. But one can show the order of convergence to be at least r.

Algorithm 3

$$\begin{bmatrix} (U_n\Lambda_n U_n^* - A)^r \\ U_n^* \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} \begin{bmatrix} R_k \\ 0 \end{bmatrix}, \text{ QR factorization}$$
$$Q_{12} = Q_1 R, \text{ QR factorization}$$
$$U_{n+1} = Q_1$$
$$\Lambda_{n+1} = U_{n+1}^* A U_{n+1}, \tag{24}$$

where Λ_1 and U_1 are $L \times L$ and $p \times L$ arbitrarily chosen. Here $1 \leq L \leq m$. One stopping criteria for Algorithms 2 and 3 would be that $||Q_{22}||$ converges to zero, where $||.||$ is any matrix norm. The matrix Λ_n converges to a matrix whose eigenvalues are the smallest (in magnitude) L eigenvalues of A.

6. Numerical Example

In this example, we illustrate how to apply Algorithm 3 to compute the matrix eigendecomposition. Consider the following 10×10 complex hermitian matrix A. To fit this matrix in a single column, it is written as $A = [A_1 : A_2]$, where A_1 and A_2 are 10×5 complex matrices representing the the first and last three columns of A, respectively. This matrix is generated using the Matlab function *rand* and scaling. The matrices A_1 and A_2 are given as:

$$A_1 = \begin{bmatrix} 0.6336 & -1.3939 & -1.7151 & -1.7836 & 0.5874 \\ -1.3939 & 2.7175 & -0.0015 & -0.3775 & 0.0962 \\ -1.7151 & -0.0015 & 2.2436 & -3.1460 & -1.1286 \\ -1.7836 & -0.3775 & -3.1460 & 2.1664 & -0.2655 \\ 0.5874 & 0.0962 & -1.1286 & -0.2655 & -3.5645 \\ 1.4382 & 1.5830 & -0.1287 & 3.2303 & -2.2926 \\ 0.3086 & -0.0036 & 0.2837 & -1.9797 & -2.6318 \\ 0.0573 & -1.0117 & -0.5414 & -2.4798 & -0.6106 \\ -1.4615 & -0.8510 & -0.7002 & -2.1118 & -0.5612 \\ -2.6896 & -0.7935 & 0.6576 & -1.9320 & 0.6709 \end{bmatrix},$$

and

$$A_2 = \begin{bmatrix} 1.4382 & 0.3086 & 0.0573 & -1.4615 & -2.6896 \\ 1.5830 & -0.0036 & -1.0117 & -0.8510 & -0.7935 \\ -0.1287 & 0.2837 & -0.5414 & -0.7002 & 0.6576 \\ 3.2303 & -1.9797 & -0.5414 & -0.7002 & 0.6576 \\ -2.2926 & -2.6318 & -0.6106 & -0.5612 & 0.6709 \\ -0.5393 & 0.0997 & 2.2295 & 2.9941 & -0.8308 \\ 0.0997 & 0.7886 & 1.4567 & 0.1901 & 0.9279 \\ 2.2295 & 1.4567 & 2.3139 & -1.7404 & 0.4480 \\ 2.9941 & 0.1901 & -1.7404 & -0.6908 & 1.7402 \\ -0.8308 & 0.9279 & 0.4480 & 1.7402 & 0.4976 \end{bmatrix}.$$

The eigenvalues of A are given by the set $\{0.2907, 1.3797, 1.6024, 3.2114, 4.2420, 5.2560, -2.9677, 8.0661, -5.7647, -8.7495\}$. Algorithm 3 is applied to compute the two smallest eigenvalues of A. It is shown that these are the eigenvalues of Λ which are $\{1.3797, 0.2907\}$. Convergence is indicated by the quantity $||Q_{22}||_2 = 7.6169(10)^{-15}$. It is also noted that $||AU_{20} - W_{20}\Lambda_{20}||_2 = 2.9299(10)^{-5}$. Here U_{20} is an approximation of invariant subspace corresponding to the smallest two eigenvalues.

6. Conclusion

In this paper, a systematic approach for generating fixed point functions of any order in rational and radical forms are proposed. These fixed point iterations can be used for computing zeros of smooth functions and eigenvalues of complex matrices. The convergence properties and numerical implementations are thoroughly analyzed. Additionally, new matrix inverse free implementations that use the QR factorization for computing an eigenvector and eigenvalue are proposed. Although these methods are only tested on polynomials and entire functions of finite number of zeros, they are also applicable to arbitrary transcendental equations including analytic and rational functions. Several important aspects concerning the numerical efficiency compared with other existing methods have not been discussed in this work. A full version of this paper is currently under consideration, where simulations and numerical evaluation of these algorithms will be established.

References

[1] P. Henrici, Applied and Computational Complex Analysis, Vol. 1, John Wiley & Sons, New York, 1974.

[2] A. S. Householder, The Numerical Treatment of a Single Nonlinear Equation, McGraw-Hill, New York, 1970.

[3] E. Schroder, "Ueber unedlich viele Algorithmen zur Auflosung der Gleichungen," Math. Ann, 2, 317-365, ξ 6.12, 1870.

[4] J. F. Traub, "A Class of Globally Convergent Iteration Functions for the Solution of Polynomial Equations," Math. Comp., 20, 113-138, $\xi\xi$ 6.9, 7.4, 1966.

[5] A. S. Householder, "Generalization of an Algorithm of Sebastiao e Silva," Numer. Math., 16, 1971, pp. 375-382.

[6] G. Polya, "Graeffe's Method for Eigenvalues," Num. Math., 11, 315-319, ξ 6.9, 1968.

[7] M. A. Hasan, "A Fast Transversal Filter for the Numerical Factorization of Polynomials," to appear in the IEEE Transactions on Circuits and Systems.

[8] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag, New York 1980.

[9] H. Maehly, "Iteration Auflosung algebraischer Gleichungen," Z. Angew. Math. Physik, 5, 260-263, 1954.

[10] G.H. Golub and C. Reinsch, "Singular Value Decomposition and Least Squares Solutions," Numer. Math. 14, pp.403-420, 1970.

[11] G. H. Golub, and C. G. Van Loan, Matrix Computations, 2nd ed., the John Hopkins University Press, Baltimore, 1989.

[12] M. A. Hasan and A. A. Hasan, "Hankel Matrices and Their Applications to the Numerical Factorization of Polynomials", Journal of Mathematical Analysis and Applications, 197, pp.459-488 (1996).

[13] M. A. Hasan and A. A. Hasan, "Hankel Matrices of Finite Rank with Applications to Signal Processing and Polynomials," J. of Math. Anal. and Appls., 208, pp.218-242, 1997.