

Extension of the Lanczos-Phillips algorithm with Laurent biorthogonal polynomials and its application to the Thron continued fractions

Kaoru Sawa¹ and Yoshimasa Nakamura¹

¹ Graduate School of Informatics, Kyoto University, Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501, Japan

E-mail sawa.kaoru.77w@st.kyoto-u.ac.jp

Received August 27, 2019, Accepted November 21, 2019

Abstract

The new Lanczos-Phillips type algorithm for computing the LDU decompositions of non-Hermitian Toeplitz matrices is presented by using Laurent biorthogonal polynomials. It can be applied to computation of the Thron continued fractions which are related to two point Padé approximations. We see that the new algorithm can compute the Thron continued fractions faster and more stably than the FG algorithm does.

Keywords Laurent biorthogonal polynomial, Lanczos-Phillips algorithm, Thron continued fraction, LDU decomposition, two point Padé approximation

Research Activity Group Applied Integrable Systems

1. Introduction

Discrete integrable systems have a deep relationship with numerical algorithms, and many researchers have studied to reveal the relationship between the two. For example, it is known that the quotient difference (qd) algorithm [1] is equivalent to the discrete Toda equation [2]. Some of recent studies aim to formulate new numerical algorithms from discrete integrable systems. The integrable singular value decomposition (I-SVD) algorithm [3, 4] is formulated from the discrete Lotka-Volterra system. Many algorithms based on discrete integrable systems have highly relative accuracy and reliability due to a positivity of solutions. Moreover, some of discrete integrable systems underlying these algorithms have Hankel determinant solutions related to orthogonal polynomials on the real line.

The Lanczos-Phillips algorithm [5] is designed by Phillips to compute the Cholesky decomposition of Hankel matrices. Generally, computing the Cholesky decomposition of a matrix needs $O(n^3)$ operations. This algorithm, however, can compute the Cholesky decomposition of a Hankel matrix within $O(n^2)$ operations. Although he derived the algorithm by using the Lanczos procedure [6], we derived, in the previous letter [7], the algorithm from a recurrence relation of orthogonal polynomials on the real line. Let $p_i(x)$ ($i = 0, 1, \dots$) be monic orthogonal polynomials on the real line of degree i . Then $p_i(x)$ ($i = 0, 1, \dots$) satisfy a orthogonality relation with a linear functional $\mathcal{L}[\cdot]$: $\mathcal{L}[p_k(x)p_l(x)] = v_l\delta_{k,l}$ ($v_l \neq 0$), and satisfy a three-term recurrence relation [8]

$$\begin{aligned} p_0(x) &= 1, \quad p_1(x) = x - a_0, \\ p_{i+1}(x) &= (x - a_i)p_i(x) - b_ip_{i-1}(x) \quad (i = 1, 2, \dots), \end{aligned} \quad (1)$$

where a_i and b_i are real numbers. Multiplying both sides of the recurrence relation (1) by x^j gives

$$x^j p_{i+1}(x) = x^{j+1} p_i(x) - a_i x^j p_i(x) - b_i x^j p_{i-1}(x).$$

Setting $r_{i,j} := \mathcal{L}[x^j p_i(x)]$, we obtain a main recurrence relation of the Lanczos-Phillips algorithm:

$$r_{i+1,j} = r_{i,j+1} - a_i r_{i,j} - b_i r_{i-1,j}.$$

Thus, the Cholesky decomposition algorithm for Hankel matrices can be regarded as being formulated from the recurrence relation of orthogonal polynomials on the real line. Other orthogonal polynomials are also expected to play key roles in designing new matrix decomposition algorithms. Based on this idea, we formulated in [7] a new Lanczos-Phillips type algorithm for computing the Cholesky decomposition of Hermitian Toeplitz matrices from a recurrence relation of orthogonal polynomials on the unit circle. Moreover, we can obtain a determinant expression of general terms of the Lanczos-Phillips algorithm easily because the orthogonal polynomials $p_i(x)$ ($i = 0, 1, \dots$) have a determinant expression. Therefore the recurrence relation of the Lanczos-Phillips algorithm is one of the discrete integrable systems in the sense of that its general terms have a Hankel determinant expression [7].

This letter is organized as follows. In Section 2, we present a new algorithm based on Laurent biorthogonal polynomials (LBPs). The algorithm can compute the LDU decompositions of non-Hermitian Toeplitz matrices. In Section 3, we show that the new Lanczos-Phillips type algorithm based on LBPs has an application to computing continued fractions. It is shown in [7] that computation of the Chebyshev continued fractions by the Lanczos-Phillips algorithm is faster and involves fewer divisions by zero than the qd algorithm.

The new algorithm based on LBPs can also compute coefficients of the Thron continued fractions (T-fractions). T-fractions are used for two point Padé approximations. We compare the new algorithm with the FG algorithm [9] in terms of computational time and the number of divisions by zero. The result indicates that our algorithm is faster and divided by zero more rarely than the FG algorithm. In Section 4, we give conclusions of this letter and discuss the future work.

2. Algorithm based on Laurent biorthogonal polynomials

Let T be a Toeplitz matrix of order n ,

$$T = [t_{j-i}]_{1 \leq i, j \leq n}. \quad (2)$$

Assume that all the leading principal minors of T are nonzero, then T is uniquely decomposed into LDU , where L is unit lower triangular, D is diagonal and U is unit upper triangular. The decomposition is called LDU decomposition. As is mentioned in the previous section, the original Lanczos-Phillips algorithm is derived from a three-term recurrence relation of orthogonal polynomials on the real line. In this section, we formulate an LDU decomposition algorithm for non-Hermitian Toeplitz matrices by using LBPs. This is main topic in this letter.

Let $\mathbb{C}[z, z^{-1}]$ be the set of whole Laurent polynomials in z , and let $\Phi_i(z)$ and $\Psi_i(z)$ ($i = 0, 1, \dots$) be monic polynomials of degree i . When $\Phi_k(z)$ and $\Psi_l(z)$ satisfy a biorthogonality relation $\mathcal{L}[\Phi_k(z)\Psi_l(z^{-1})] = w_l \delta_{k,l}$ ($w_l \neq 0$) with a linear functional $\mathcal{L} : \mathbb{C}[z, z^{-1}] \rightarrow \mathbb{C}$, then $\Phi_i(z)$ and $\Psi_i(z)$ ($i = 0, 1, \dots$) are Laurent biorthogonal polynomials [9]. Moreover, $t_k, k \in \mathbb{Z}$ are defined by $t_k := \mathcal{L}[z^k]$, then $\Phi_i(z)$ and $\Psi_i(z)$ are expressed as

$$\begin{aligned} \Phi_i(z) &= \frac{1}{T_i} \begin{vmatrix} t_0 & t_1 & \dots & t_i \\ t_{-1} & t_0 & \dots & t_{i-1} \\ \vdots & \vdots & & \vdots \\ t_{-i+1} & t_{-i+2} & \dots & t_1 \\ 1 & z & \dots & z^i \end{vmatrix}, \\ \Psi_i(z) &= \frac{1}{T_i} \begin{vmatrix} t_0 & t_{-1} & \dots & t_{-i} \\ t_1 & t_0 & \dots & t_{-i+1} \\ \vdots & \vdots & & \vdots \\ t_{i-1} & t_{i-2} & \dots & t_{-1} \\ 1 & z^{-1} & \dots & z^{-i} \end{vmatrix}, \\ T_0 &= 1, \quad T_j = |t_{l-k}|_{1 \leq k, l \leq j}. \end{aligned} \quad (3)$$

When $t_k = t_{-k}$: $\Phi_i(z) = \Psi_i(z)$, the LBPs are called “orthogonal polynomials on the unit circle” [10]. The set of orthogonal polynomials on the unit circle is a reduction of that of LBPs, and we derive in [7] the Cholesky decomposition algorithm for Hermitian Toeplitz matrices by orthogonal polynomials on the unit circle.

As with orthogonal polynomials on the real line, LBPs satisfy the following three-term recurrence relations

$$\begin{aligned} \Phi_{i+1}(z) &= (z + c_i)\Phi_i(z) - d_i z \Phi_{i-1}(z), \\ \Psi_{i+1}(z^{-1}) &= (z^{-1} + e_i)\Psi_i(z^{-1}) - f_i z^{-1} \Psi_{i-1}(z^{-1}), \end{aligned} \quad (4)$$

$$(5)$$

where

$$c_i = -\frac{T_i \hat{T}_{i+1}}{T_{i+1} \hat{T}_i}, \quad d_i = -\frac{T_{i-1} \hat{T}_{i+1}}{T_i \hat{T}_i}, \quad (6)$$

$$e_i = -\frac{T_i \tilde{T}_{i+1}}{T_{i+1} \tilde{T}_i}, \quad f_i = -\frac{T_{i-1} \tilde{T}_{i+1}}{T_i \tilde{T}_i}, \quad (7)$$

$$(\hat{T}_i = |t_{l-k+1}|_{1 \leq k, l \leq i}, \quad \tilde{T}_i = |t_{l-k-1}|_{1 \leq k, l \leq i}).$$

Multiplying both sides of the recurrence relation (4) by z^{-j} and (5) by z^j gives

$$z^{-j} \Phi_{i+1}(z) = (z + c_i) z^{-j} \Phi_i(z) - d_i z^{-j+1} \Phi_{i-1}(z),$$

$$z^j \Psi_{i+1}(z^{-1}) = (z^{-1} + e_i) z^j \Psi_i(z^{-1}) - f_i z^{j-1} \Psi_{i-1}(z^{-1}).$$

Setting $l_{i,j} := \mathcal{L}[z^{-j} \Phi_i(z)]$ and $u_{i,j} := \mathcal{L}[z^j \Psi_i(z^{-1})]$, we obtain recurrence relations

$$\begin{cases} l_{i+1,j} = l_{i,j-1} + c_i l_{i,j} - d_i l_{i-1,j-1} \\ u_{i+1,j} = u_{i,j-1} + e_i u_{i,j} - f_i u_{i-1,j-1} \end{cases}. \quad (8)$$

We can obtain the general term of recurrence relations (8) by the determinant expression of LBPs (3):

$$l_{i,j} = \frac{T_{i,j}}{T_{i-1,i-1}}, \quad u_{i,j} = \frac{\bar{T}_{i,j}}{\bar{T}_{i-1,i-1}}, \quad (9)$$

where $T_{-1,-1} := 1$, $T_{0,j} := t_{-j}$, $\bar{T}_{-1,-1} := 1$, $\bar{T}_{0,j} := t_j$,

$$\begin{aligned} T_{i,j} &:= \begin{vmatrix} t_0 & t_1 & \dots & t_i \\ t_{-1} & t_0 & \dots & t_{i-1} \\ \vdots & \vdots & & \vdots \\ t_{-i+1} & t_{-i+2} & \dots & t_1 \\ t_{-j} & t_{-j+1} & \dots & t_{-j+i} \end{vmatrix}, \\ \bar{T}_{i,j} &:= \begin{vmatrix} t_0 & t_{-1} & \dots & t_{-i} \\ t_1 & t_0 & \dots & t_{-i+1} \\ \vdots & \vdots & & \vdots \\ t_{i-1} & t_{i-2} & \dots & t_{-1} \\ t_j & t_{j-1} & \dots & t_{j-i} \end{vmatrix}. \end{aligned}$$

We show that the LDU decomposition can be computed by the recurrence relations (9). Let A be a matrix of order n . If $A = [a_{i,j}]_{1 \leq i, j \leq n}$ has the LDU decomposition $A = L_A D_A U_A$, then elements of L_A , D_A and U_A are expressed by some determinants:

$$(L_A)_{i,j} = \frac{A_{j,i}}{A_{j,j}}, \quad (D_A)_{i,i} = \frac{A_{i,i}}{A_{i-1,i-1}}, \quad (U_A)_{i,j} = \frac{\bar{A}_{i,j}}{\bar{A}_{i,i}}, \quad (10)$$

where $A_{0,0} := 1$, $A_{1,j} := a_{j,1}$, $\bar{A}_{0,0} := 1$, $\bar{A}_{1,j} := a_{1,j}$,

$$\begin{aligned} A_{i,j} &:= \begin{vmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,i} \\ a_{2,1} & a_{2,2} & \dots & a_{2,i} \\ \vdots & \vdots & & \vdots \\ a_{i-1,1} & a_{i-1,2} & \dots & a_{i-1,i} \\ a_{j,1} & a_{j,2} & \dots & a_{j,i} \end{vmatrix}, \\ \bar{A}_{i,j} &:= \begin{vmatrix} a_{1,1} & a_{2,1} & \dots & a_{i,1} \\ a_{1,2} & a_{2,2} & \dots & a_{i,2} \\ \vdots & \vdots & & \vdots \\ a_{1,i-1} & a_{2,i-1} & \dots & a_{i,i-1} \\ a_{1,j} & a_{2,j} & \dots & a_{i,j} \end{vmatrix}. \end{aligned}$$

Using the determinant expression (9) and the determinant expression of general LDU decompositions (10), if the Toeplitz matrix (2) can be decomposed as $T = LDU$, the following equality holds

$$(L)_{i,j} = \frac{l_{j-1,i-1}}{l_{j-1,j-1}}, \quad (D)_{i,i} = l_{i-1,i-1}, \quad (U)_{i,j} = \frac{u_{i-1,j-1}}{u_{i-1,i-1}}.$$

Therefore we can compute LDU decompositions of Toeplitz matrices using the recurrence relations (8). Using determinant expressions (6) and (7), auxiliary variables c_i, d_i, e_i, f_i are represented by $l_{i,j}$ and $u_{i,j}$, namely,

$$c_i = \frac{l_{i-1,i-1}l_{i,-1}}{l_{i,i}l_{i-1,-1}}, \quad d_i = \frac{l_{i,-1}}{l_{i-1,-1}},$$

$$e_i = \frac{u_{i-1,i-1}u_{i,-1}}{u_{i,i}u_{i-1,-1}}, \quad f_i = \frac{u_{i,-1}}{u_{i-1,-1}}$$

because $\hat{T}_i = (-1)^{i-1}T_{i-1,-1}$ and $\tilde{T}_i = (-1)^{i-1}\bar{T}_{i-1,-1}$. Furthermore, initial values of the recurrence relations (8) are determined as follows

$$l_{0,j} = t_{-j}, \quad c_0 = \frac{l_{0,-1}}{l_{0,0}}, \quad d_0 = 0,$$

$$u_{0,j} = t_j, \quad e_0 = \frac{u_{0,-1}}{u_{0,0}}, \quad f_0 = 0.$$

In summary, we obtain the following algorithm.

Algorithm 1 (LDU decomposition algorithm based on LBP's)

0) For the Toeplitz matrix (2), set

$$l_{0,j} = t_{-j}, \quad u_{0,j} = t_j \quad (j = -n+1, \dots, n-1),$$

$$c_0 = -\frac{l_{0,-1}}{l_{0,0}}, \quad d_0 = 0, \quad e_0 = -\frac{u_{0,-1}}{u_{0,0}}, \quad f_0 = 0.$$

1) Compute repeatedly from $i = 1$ to $n-1$

$$l_{i,i} = u_{i,i} = l_{i-1,i-1} + c_{i-1}l_{i-1,i-1} - d_{i-1}l_{i-2,i-2},$$

$$l_{i,j} = l_{i-1,j-1} + c_{i-1}l_{i-1,j} - d_{i-1}l_{i-2,j-1},$$

$$u_{i,j} = u_{i-1,j-1} + e_{i-1}u_{i-1,j} - f_{i-1}u_{i-2,j-1}$$

$$(j = -n+i+1, \dots, -1, i+1, i+2, \dots, n-1),$$

$$d_i = \frac{l_{i,-1}}{l_{i-1,-1}}, \quad c_i = d_i \frac{l_{i-1,i-1}}{l_{i,i}} \quad (i \neq n-1),$$

$$f_i = \frac{u_{i,-1}}{u_{i-1,-1}}, \quad e_i = f_i \frac{u_{i-1,i-1}}{u_{i,i}} \quad (i \neq n-1).$$

2) Set

$$(D)_{i,i} = l_{i-1,i-1} \quad (i = 1, \dots, n),$$

$$(L)_{i,j} = \delta_{i,j} \quad (i \leq j), \quad (L)_{i,j} = \frac{l_{j-1,i-1}}{l_{j-1,j-1}} \quad (i > j),$$

$$(U)_{i,j} = \delta_{i,j} \quad (i \geq j), \quad (U)_{i,j} = \frac{u_{i-1,j-1}}{u_{i-1,i-1}} \quad (i < j).$$

Generally, matrices have $O(n^2)$ degrees of freedom, and their LDU decompositions can be computed within $O(n^3)$ operations. However, Toeplitz matrices have only $O(n)$ degree of freedom which is, in fact, just suitable in relating to orthogonal polynomials, hence this algorithm can compute their LDU decompositions within $O(n^2)$ operations.

Numerical example 1

Let us consider the following matrix:

$$T = \begin{bmatrix} 7 & 6 & 4 \\ 5 & 7 & 6 \\ 3 & 5 & 7 \end{bmatrix}.$$

The first step is to set the initial values

$$l_{0,-2} = u_{0,2} = 4, \quad l_{0,-1} = u_{0,1} = 6, \quad l_{0,0} = u_{0,0} = 7,$$

$$l_{0,1} = u_{0,-1} = 5, \quad l_{0,2} = u_{0,-2} = 3, \quad d_0 = f_0 = 0,$$

$$c_0 = -\frac{l_{0,-1}}{l_{0,0}} = -\frac{6}{7}, \quad e_0 = -\frac{u_{0,-1}}{u_{0,0}} = -\frac{5}{7}.$$

When $i = 1$,

$$l_{1,-1} = l_{0,-2} + c_0 l_{0,-1} = -\frac{8}{7}, \quad l_{1,1} = l_{0,0} + c_0 l_{0,1} = \frac{19}{7},$$

$$l_{1,2} = l_{0,1} + c_0 l_{0,2} = \frac{17}{7}, \quad d_1 = \frac{l_{1,-1}}{l_{0,-1}} = -\frac{4}{21},$$

$$c_1 = \frac{d_1 l_{0,0}}{l_{1,1}} = -\frac{28}{57}, \quad u_{1,-1} = u_{0,-2} + e_0 u_{0,-1} = -\frac{4}{7},$$

$$u_{1,1} = u_{0,0} + e_0 u_{0,1} = \frac{19}{7}, \quad u_{1,2} = u_{0,1} + e_0 u_{0,2} = \frac{22}{7},$$

$$f_1 = \frac{u_{1,-1}}{u_{0,-1}} = -\frac{4}{35}, \quad e_1 = \frac{f_1 u_{0,0}}{u_{1,1}} = -\frac{28}{95}.$$

When $i = 2$,

$$l_{2,2} = u_{2,2} = l_{1,1} + c_1 l_{1,2} - d_1 l_{0,1} = \frac{47}{19}.$$

Therefore T is decomposed as $T = LDU$:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ \frac{5}{7} & 1 & 0 \\ \frac{3}{7} & \frac{17}{19} & 1 \end{bmatrix} \begin{bmatrix} 7 & 0 & 0 \\ 0 & \frac{19}{7} & 0 \\ 0 & 0 & \frac{47}{19} \end{bmatrix} \begin{bmatrix} 1 & \frac{6}{7} & \frac{4}{7} \\ 0 & 1 & \frac{22}{19} \\ 0 & 0 & 1 \end{bmatrix}.$$

3. Application to the T-fractions

The original Lanczos-Phillips algorithm can compute the Chebyshev continued fractions. The algorithm is faster and involves fewer zero divisions than the qd algorithm, which can compute the Chebyshev continued fractions. Moreover, the Lanczos-Phillips type algorithm based on orthogonal polynomials on the unit circle can compute the Perron continued fractions and it is faster than the discrete Schur flow, which can compute the Perron continued fractions [7]. Algorithm 1 also has an application to continued fractions.

Let $G(z)$ be a complex function. If $G(z)$ is expressed as expansion at zero and infinity

$$\begin{cases} G(z) = -t_1 z - t_2 z^2 - t_3 z^3 - \dots \\ G(z) = t_0 + \frac{t_{-1}}{z} + \frac{t_{-2}}{z^2} + \dots \end{cases}, \quad (11)$$

and the coefficients $t_{-n+1}, \dots, t_{-1}, t_0, t_1, \dots, t_n$ are already obtained, then $G(z)$ can be approximated by the Thron continued fraction (T-fraction)

$$\hat{G}(z) = \frac{-t_1 z}{1 + c_0 z - \frac{d_1 z}{1 + c_1 z - \dots - \frac{d_{n-1} z}{1 + c_{n-1} z}}}, \quad (12)$$

$$c_i = -\frac{T_i \hat{T}_{i+1}}{T_{i+1} \hat{T}_i}, \quad d_i = -\frac{T_{i-1} \hat{T}_{i+1}}{T_i \hat{T}_i}.$$

The rational function $\hat{G}(z)$ gives a two point Padé approximation of $G(z)$. Because c_i and d_i have the same

Table 1. Computation environment condition.

CPU	1.7 GHz Intel Core i7
Memory	8 GB 1600 MHz DDR3
OS	Linux Ubuntu (ver 16.04)
Programing language	C++

Table 2. Computational time (sec.).

Size n	Algorithm 1	FG algorithm
10^3	0.00477	0.0117
5×10^3	0.131	0.293
10^4	0.577	1.13

Table 3. Percentage of zero divisions (%).

Algorithm 1	FG algorithm
0.007	3.814

determinant expression as (6), Algorithm 1 can compute the coefficients of the T-fraction (12). In this case, the coefficients of the expansion (11) are initial values for the computing. We need to compute c_{n-1} and d_{n-1} in Algorithm 1, while we don't need $u_{i,j}$, e_i and f_i .

It is known that the FG algorithm can also compute the coefficients of the T-fraction for $G(z)$ [9]. The FG algorithm is illustrated as follows.

Algorithm 2 (*FG algorithm*)

0) When the coefficients $t_{-n+1}, \dots, t_0, \dots, t_n$ of $G(z)$ (11) are given, set

$$F_j^{(0)} = 0, \quad G_j^{(0)} = -\frac{t_{j+1}}{t_j} \quad (j = -n+1, \dots, n-1).$$

1) Compute repeatedly from $i = 0$ to $n - 2$

$$F_j^{(i+1)} = F_{j+1}^{(i)} + G_{j+1}^{(i)} - G_j^{(i)} \\ (j = -n + i + 1, \dots, 0, \dots, n - i - 2),$$

$$G_j^{(i+1)} = \frac{F_j^{(i+1)}}{F_{j-1}^{(i+1)}} G_{j-1}^{(i)}$$

$$(j = -n + i + 2, \dots, 0, \dots, n - i - 2).$$

2) Set

$$c_0 = G_0^{(0)}, \quad c_i = G_0^{(i)}, \\ d_i = -F_0^{(i)} \quad (i = 1, \dots, n - 1).$$

First we compare Algorithm 1 with the FG algorithm in terms of computational time. Randomly generated numbers t_j in the range $(0, 1]$ are given to Algorithms 1 and the FG algorithm as their initial values. Table 1 shows our computation environment condition. Table 2 is the result of experiments. Algorithm 1 can compute faster than the FG algorithm. This is because Algorithm 1 has fewer divisions than the FG algorithm. Although the FG algorithm has $O(n^2)$ divisions, Algorithm 1 has $O(n)$ divisions. Generally, division has the largest time complexity among the four arithmetic operations.

Next, let us compare two algorithms in terms of reliability. Because Algorithm 1 has fewer divisions, it may behave well even in a near-breakdown situation. Randomly generated 10^3 numbers in the range $(0, 1]$ are

given to each algorithm, and we count the number of zero divisions in 10^6 trials. Table 3 is the result of experiments. The percentages are the number of zero divisions divided by the number of trials. Fewer zero divisions occur in Algorithm 1 than those in the FG algorithm. This result suggests that Algorithm 1 is a more reliable numerical algorithm.

4. Conclusions

In this letter, we obtain a new algorithm for LDU decompositions of non-Hermitian Toeplitz matrices (Algorithm 1). Moreover, we show that Algorithm 1 can compute the T-fractions and it is faster and involves fewer zero divisions than the FG algorithm. Algorithm 1 can be regarded as an extension of the Lanczos-Phillips algorithm to the LBPs. Considering that many discrete integrable systems have Hankel or Toeplitz determinant solutions, we regard the recurrence relation of the Lanczos-Phillips algorithm and Algorithm 1 as discrete integrable systems in the sense in which the general term of their recurrence relations has some determinant expressions. In that case, it is expected that the Lanczos-Phillips type algorithm is a numerical algorithm which has a better reliability than the existing algorithms.

We plan, in the future work, to associate other orthogonal polynomials with matrix decompositions along the line similar to formulating Algorithm 1 to other orthogonal polynomials and reveal unexpected relationships between the Lanczos-Phillips type algorithms and discrete integrable systems.

Acknowledgments

The authors would like to thank the anonymous referee who reviewed the earlier version of this manuscript and provided valuable suggestions and comments.

References

- [1] H. Rutishauser, Der Quotienten-Differenzen-Algorithmus, Z. Angew. Math. Phys., **5** (1954), 233–251.
- [2] Y. Nakamura, K. Takasaki, S. Tsujimoto, M. Okado and J. Inoguchi, Kaiseigaku Hyakka II Kaseibunkei no Suuri (in Japanese), Asakura Shoten, Tokyo, 2018.
- [3] M. Iwasaki, Studies of Singular Value Decomposition in Terms of Integrable Systems, Ph.D. Thesis, Kyoto University, 2004.
- [4] M. Iwasaki, S. Sakano and Y. Nakamura, Accurate twisted factorization of real symmetric tridiagonal matrices and its application to singular value decomposition (in Japanese), Trans. JSIAM, **15** (2005), 461–481.
- [5] J. L. Phillips, The triangular decomposition of Hankel matrices, Math. Comp., **25** (1971), 599–602.
- [6] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, J. Res. Natl. Bur. Stand., **45** (1950), 255–282.
- [7] K. Sawa and Y. Nakamura, Application of the Lanczos-Phillips algorithm to continued fractions and its extension with orthogonal polynomials, JSIAM Lett., **10** (2018), 57–60.
- [8] T. S. Chihara, An Introduction to Orthogonal Polynomials, Gordon and Breach Science Publishers, New York, 1978.
- [9] W. B. Jones and W. J. Thron, Continued Fractions: Analytic Theory and Applications, Addison-Wesley Pub. Co., Reading, 1980.
- [10] N. I. Akhiezer, The Classical Moment Problem: and Some Related Questions in Analysis, Olver & Boyd, Edinburgh, 1965.