JSIAM Letters Vol.10 (2018) pp.57–60 ©2018 Japan Society for Industrial and Applied Mathematics

Application of the Lanczos-Phillips algorithm to continued fractions and its extension with orthogonal polynomials

Kaoru Sawa[1] and Yoshimasa Nakamura[2]

[1] Faculty of Engineering, Kyoto University, 36-1 Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
[2] Graduate School of Informatics, Kyoto University, 36-1 Yoshida Honmachi, Sakyo-ku, Kyoto 606-8501, Japan

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

Received June 13, 2018, Accepted August 9, 2018

Abstract

The Lanczos-Phillips algorithm is a method to compute the Cholesky decomposition of Hankel matrices. This algorithm can compute the Chebyshev continued fraction from a given set of moments faster than the qd algorithm. Moreover, a new algorithm to compute the Perron continued fraction is presented with the help of orthogonal polynomials. The Cholesky decomposition of Toeplitz matrices plays a key role.

Keywords continued fraction, orthogonal polynomial, Padé approximation, Lanczos-Philips algorithm, Schur algorithm

Research Activity Group Applied Integrable Systems

1. Introduction

Recent studies have aimed to formulate numerical algorithms based on discrete integrable systems. Examples of such algorithms include the dLVs algorithm (discrete Lotka-Volterra equations) and the dqds algorithm (discrete Toda equations) [1]. These algorithms have high relative accuracies. Moreover, these equations have Hankel determinant solutions and are closely related to orthogonal polynomials.

On the other hand, Phillips designed the Lanczos-Phillips algorithm to decompose a Hankel matrix within $O(n^2)$ operations over 45 years ago [2]. He derived the algorithm using the Lanczos procedure and showed that it can compute the coefficients of three-term recurrence relations for orthogonal polynomials on a real line.

This letter has three purposes. Firstly, we present a Hankel determinant expression for the general terms of the recurrence relation in the Lanczos-Phillips algorithm. This expression can be derived from orthogonal polynomials on a real line.

Secondly, we show that the Lanczos-Phillips algorithm can be applied to continued fractions. Deriving a continued fraction expansion of an analytic function given by a power series is a fundamental problem of applied mathematics. A continued fraction expansion, which is closely related to the Padé approximation, has applications to control theory and stability theory. The most standard continued fraction is the Chebyshev type [3]. It is known that the qd algorithm by Rutishauser [4] can compute the Chebyshev continued fraction [3]. A comparison between the Lanczos-Phillips algorithm and the qd algorithm demonstrates that the former is faster and has a better numerical reliability.

Finally, by using a method to apply the Lanczos-Phillips algorithm to continued fractions, we design a new algorithm with the help of orthogonal polynomials on a unit circle. This new algorithm can compute coefficients of the Perron continued fraction. It is known that the discrete Schur flow by Mukaihira and Nakamura [5] can also compute the Perron continued fraction. A comparison between of these two algorithms confirms that our algorithm has a faster computation time than the discrete Schur flow.

2. Determinant expression of the general terms of the Lanczos-Phillips algorithm

Let H be a Hankel matrix of order n,

$$H = [h_{i+j-2}]_{1\le i,j\le n}, \tag{1}$$

where h_k are some real numbers. Assume that all the leading principal minors of H are nonzero. Then H is uniquely decomposed into $R^\top DR$, where R is a unit upper triangular matrix and D is a diagonal matrix. Here, the symbol '$\top$' denotes a transposition. The formula $H = R^\top DR$ is usually called the Cholesky decomposition. The Cholesky decomposition of H is performed within $O(n^2)$ operations using the Lanczos-Phillips algorithm [2]. This algorithm designed from the Lanczos procedure is sketched as follows.

Algorithm 1 *(Lanczos-Phillips algorithm)*

0) For the Hankel matrix (1)*, set*

$$c_{1,j} = h_{j-1}, \quad j = 1, 2, \ldots, 2n-1,$$

$$a_1 = \frac{c_{1,2}}{c_{1,1}}, \quad b_1 = 0.$$

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

$$c_{i+1,j} = c_{i,j+1} - a_i c_{i,j} - b_i c_{i-1,j}, \tag{2}$$

for $j = i+1, i+2, \ldots, 2n-i-1$,
and if $i \neq n-1$,

$$a_{i+1} = \frac{c_{i+1,i+2}}{c_{i+1,i+1}} - \frac{c_{i,i+1}}{c_{i,i}}, \quad b_{i+1} = \frac{c_{i+1,i+1}}{c_{i,i}}.$$

2) Set $(DR)_{i,j} = c_{i,j}$, $i = 1, 2, \ldots, n$; $j = i, i+1 \ldots, n$. Then R is a unit upper triangular matrix and D is a diagonal matrix. Thus, rows of R are DR divided by the element of D.

In general, the Cholesky decomposition is computed within $O(n^3)$ operations, but the Lanczos-Phillips algorithm can decompose this class of the Hankel matrices within $O(n^2)$ operations.

The auxiliary variables a_i and b_i computed by the algorithm have another significance. Actually, they are coefficients for the three-term recurrence relations of orthogonal polynomials on a real line.

$$\begin{aligned} &p_0(x) = 1, \quad p_1(x) = x - a_1, \\ &p_i(x) = (x - a_i)p_{i-1}(x) - b_i p_{i-2}(x), \\ &i = 2, 3, \ldots, n. \end{aligned} \tag{3}$$

$\{p_i(x)\}_{i=0}^n$ is called the Lanczos polynomial and is determined by the moments $h_0, h_1, \ldots, h_{2n-2}$, namely, the elements of H [6]. $\{p_i(x)\}_{i=0}^n$ is orthogonal with respect to a linear functional $\mathcal{L}[\cdot]$, which defines the moments as

$$\mathcal{L}[x^j] = h_j, \quad j = 0, 1, \ldots, 2n-2. \tag{4}$$

Moreover, $\{p_i(x)\}_{i=0}^n$ has a determinant expression with the moments [7]

$$p_i(x) = \frac{1}{H_i} \begin{vmatrix} h_0 & h_1 & \ldots & h_i \\ h_1 & h_2 & \ldots & h_{i+1} \\ \vdots & \vdots & & \vdots \\ h_{i-1} & h_i & \ldots & h_{2i-1} \\ 1 & x & \ldots & x^i \end{vmatrix}, \tag{5}$$

$$H_0 = 1, \quad H_i = |h_{k+l-2}|_{1 \le k,l \le i}, \quad i = 0, 1, \ldots, n-1.$$

Multiplying both sides of the recurrence relation (3) by x^j gives

$$x^j p_i(x) = x^{j+1} p_i(x) - a_i x^j p_{i-1}(x) - b_i x^j p_{i-2}(x).$$

Let $H_{i,j}$ be a determinant in the form

$$H_{-1,-1} := 1, \quad H_{0,j} := h_j,$$

$$H_{i,j} := \begin{vmatrix} h_0 & h_1 & \ldots & h_i \\ h_1 & h_2 & \ldots & h_{i+1} \\ \vdots & \vdots & & \vdots \\ h_{i-1} & h_i & \ldots & h_{2i-1} \\ h_j & h_{j+1} & \ldots & h_{j+i} \end{vmatrix}.$$

Since $\{p_i(x)\}_{i=0}^n$ has a determinant expression (5), $\mathcal{L}[x^j p_i(x)]$ is expressed as

$$\mathcal{L}[x^j p_i(x)] = \frac{H_{i,j}}{H_{i-1,i-1}}.$$

Table 1. Environment condition.

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

If $c_{i,j} = \mathcal{L}[x^{j-1} p_{i-1}(x)]$, then $\{c_{i,j}\}$ satisfies the recurrence relation (2). Therefore, we obtain the following theorem.

Theorem 1 *The recurrence relation* (2) *has a determinant expression of*

$$c_{i+1,j+1} = \frac{H_{i,j}}{H_{i-1,i-1}},$$

$$i = 0, 1, \ldots, n-1, \quad j = 0, 1, \ldots, 2n-i-2.$$

3. Application to a continued fraction

Let $f(x)$ be the formal power series such that

$$f(x) = \frac{h_0}{x} + \frac{h_1}{x^2} + \cdots + \frac{h_n}{x^{n+1}} + \ldots. \tag{6}$$

It is known [3] that $f(x)$ can be expressed by the Chebyshev continued fraction

$$f(x) = \cfrac{h_0}{x - a_1 - \cfrac{b_2}{x - a_2 - \cfrac{b_3}{x - a_3 - \ddots}}}, \tag{7}$$

and a_i and b_i are

$$a_i = \frac{H_{i,i+1}}{H_{i,i}} - \frac{H_{i-1,i}}{H_{i-1,i-1}}, \quad b_i = \frac{H_{i,i} H_{i-2}}{{H_{i-1,i-1}}^2}. \tag{8}$$

Because a_i and b_i are written by (8), the Lanczos-Phillips algorithm can compute the coefficients for the continued fraction (7) of the power series (6) through Theorem 1 concerning a determinant structure of the algorithm. Consider the case where a finite number of data, $h_0, h_1, \ldots, h_{2n-1}$, are given. We can obtain $a_1, a_2, \ldots, a_n, b_2, b_3, \ldots, b_n$ by the algorithm. Setting $b_{n+1} = 0$ yields a truncated continued fraction $\hat{f}(x)$. The power series expansion of $\hat{f}(x)$ matches $f(x)$ from h_0/x to h_{2n-1}/x^{2n}. Therefore, $\hat{f}(x)$ is a Padé approximation of the given function $f(x)$.

Rutishauser's qd algorithm can also compute the coefficients for the continued fraction (7) of the power series (6) [3]. Thus, let us compare the computation time between the Lanczos-Phillips algorithm and the qd algorithm. Randomly generated $10^3, 5 \times 10^3$, and 10^4 numbers in the range $[0, 1]$ are given as moments h_k. Table 1 presents the environment conditions. Table 2 shows that the Lanczos-Phillips algorithm has a superior (shorter) computation time than the qd algorithm. This is because the Lanczos-Phillips algorithm has fewer divisions than the qd algorithm. In fact, the qd algorithm has $O(n^2)$ divisions [3], whereas the Lanczos-Phillips algorithm (Algorithm 1) has $O(n)$ divisions. Generally, division has the largest time complexity among the four arithmetic operations.

Next, let us compare the two algorithms in terms of

Table 2. Computation time (sec.).

Number of moments	Lanczos-Phillips alg.	qd alg.
10^3	0.00758	0.0183
5×10^3	0.172	1.01
10^4	0.651	8.80

reliability. The possibility of numerical instability of the qd algorithm is reported in [3] when the denominator is nearly equal to zero. Because the Lanczos-Phillips algorithm has fewer divisions, it should behave well even in a near-breakdown situation. Randomly generated 10^3 numbers in the range $[0, 1]$ are given as the moments (4), respectively. We count the number of failures in 10^5 trials. Table 3 shows that the Lanczos-Phillips algorithm has a better reliability for the same set of moments.

4. Algorithm based on orthogonal polynomials on a unit circle

Next let us consider an extension of the Lanczos-Phillips algorithm to a Toepliz matrix and orthogonal polynomials on a unit circle. Let $\mathbb{D} = \{z \in \mathbb{C} | \ |z| = 1\}$ be a unit circle, where $z = \mathrm{e}^{\mathrm{i}\theta}$ for $0 \le \theta \le 2\pi$ and $\mathrm{i} = \sqrt{-1}$. Moreover, let $\sigma(\theta)$ be a measure on the unit circle $\mathbb{D}$. The inner product on $\mathbb{D}$ is defined by

$$\langle f(z), g(z)\rangle = \int_0^{2\pi} \overline{f(z)} g(z) \mathrm{d}\sigma(\theta), \quad z = \mathrm{e}^{\mathrm{i}\theta}. \tag{9}$$

Let $\Phi_n(z), n = 0, 1, \ldots$ be monic polynomials of degree n. If $\{\Phi_n(z)\}$ is orthogonal with respect to the inner product (9),

$$\langle \Phi_i(z), \Phi_j(z)\rangle = d_j \delta_{i,j}, \quad d_j \neq 0,$$

then $\{\Phi_n(z)\}$ is called a orthogonal polynomial on a unit circle [8]. Let $\delta_{i,j}$ denote the Kronecker delta. Let t_n be the moments defined by

$$t_n = \int_0^{2\pi} \mathrm{e}^{\mathrm{i}n\theta} \mathrm{d}\sigma(\theta), \quad n \in \mathbb{Z}.$$

The polynomial $\Phi_n(z)$ can also be expressed as a determinant form with the moments such that

$$\Phi_n(z) = \frac{1}{T_n} \begin{vmatrix} t_0 & t_1 & \ldots & t_n \\ t_{-1} & t_0 & \ldots & t_{n-1} \\ \vdots & \vdots & & \vdots \\ t_{-n+1} & t_{-n+2} & \ldots & t_1 \\ 1 & z & \ldots & z^n \end{vmatrix},$$

$$T_0 = 1, \quad T_n = |t_{i-j}|_{1 \le i,j \le n}.$$

Note that $t_{-n} = \overline{t_n}$ because $\overline{\mathrm{e}^{in\theta}} = \mathrm{e}^{-in\theta}$. Monic orthogonal polynomials on a unit circle can also satisfy the following three-term recurrence relations

$$\Phi_{i+1}(z) = z\Phi_i(z) + u_i \Phi_i(z) - v_i z \Phi_{i-1}(z), \tag{10}$$

$$u_i = -\frac{T_i \hat{T}_{i+1}}{T_{i+1}\hat{T}_i}, \quad v_i = -\frac{T_{i-1}\hat{T}_{i+1}}{T_i \hat{T}_i}, \tag{11}$$

$$\hat{T}_i = |t_{n-m+1}|_{1 \le n,m \le i}.$$

Table 3. Failure percentage (%).

Lanczos-Phillips algorithm	qd algorithm
0	3.41

As is discussed in Sections 2 and 3, the Lanczos-Phillips algorithm is derived from the three-term recurrence relations of orthogonal polynomials on a real line. Now, let us consider a new algorithm based on the orthogonal polynomials on a unit circle. There is a difference between the recurrence relations (3) and (10).

Multiplying both sides of the recurrence relation (10) by z^{-j} gives

$$z^{-j}\Phi_{i+1}(z) = z^{-j+1}\Phi_i(z) + u_i z^{-j}\Phi_i(z) - v_i z^{-j+1}\Phi_{i-1}(z).$$

Let $T_{i,j}$ be a determinant such that

$$T_{-1,-1} := 1, \quad T_{0,j} := t_{-j},$$

$$T_{i,j} := \begin{vmatrix} t_0 & t_1 & \ldots & t_i \\ t_{-1} & t_0 & \ldots & t_{i-1} \\ \vdots & \vdots & & \vdots \\ t_{-i+1} & t_{-i+2} & \ldots & t_1 \\ t_{-j} & t_{-j+1} & \ldots & t_{-j+i} \end{vmatrix}. \tag{12}$$

If we set $s_{i,j} := \langle 1, z^{-j}\Phi_i(z)\rangle$, then

$$s_{i,j} = \frac{T_{i,j}}{T_{i-1,i-1}}. \tag{13}$$

Therefore, we obtain a recurrence relation

$$s_{i+1,j} = s_{i,j-1} + u_i s_{i,j} - v_i s_{i-1,j-1}. \tag{14}$$

If the auxiliary variables u_i and v_i are determined from $s_{i,j}$, all $s_{i,j}$ can be computed with the recurrence relation (14). u_i and v_i have the determinant expressions (11). Now put $\hat{T}_i = (-1)^{i-1} T_{i-1,-1}$. Then u_i and v_i are expressed with $s_{i,j}$ as follows

$$u_i = -\frac{T_i \hat{T}_{i+1}}{T_{i+1}\hat{T}_i} = \frac{s_{i-1,i-1}s_{i,-1}}{s_{i,i}s_{i-1,-1}},$$

$$v_i = -\frac{T_{i-1}\hat{T}_{i+1}}{T_i \hat{T}_i} = \frac{s_{i,-1}}{s_{i-1,-1}}.$$

Finally, let us fix the initial values of the recurrence relation. Using (11) and (12), we have

$$s_{0,j} = \frac{T_{0,j}}{T_{-1,-1}} = t_{-j}, \quad v_0 = -\frac{T_{-1}\hat{T}_1}{T_0\hat{T}_0} = 0,$$

$$u_0 = -\frac{T_0\hat{T}_1}{T_1\hat{T}_0} = -\frac{t_1}{t_0} = -\frac{s_{0,-1}}{s_{0,0}}.$$

In summary, we obtain

Algorithm 2 *(An algorithm associated with orthogonal polynomials on a unit circle)*

0) If the moments $\{t_j\}_{-n+1 \le j \le n-1}$ satisfy $t_{-n} = \overline{t_n}$, are given such that $T_{i,i} \neq 0$, then set

$$s_{0,j} = t_{-j}, \quad j = -n+1, \ldots, n-1,$$

$$u_0 = -\frac{s_{0,-1}}{s_{0,0}}, \quad v_0 = 0.$$

Table 4. Computation time (sec.).

Number of moments	Algorithm 2	discrete Schur flow
10^3	0.00975	0.0151
5×10^3	0.172	0.409
10^4	0.671	1.84

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

$$s_{i+1,j} = s_{i,j-1} + u_i s_{i,j} - v_i s_{i-1,j-1},$$
$$j = -n + i + 1, \ldots, -1, i, i + 1, \ldots, n - 1,$$
$$v_{i+1} = \frac{s_{i+1,-1}}{s_{i,-1}}, \quad u_{i+1} = v_{i+1} \frac{s_{i,i}}{s_{i+1,i+1}}.$$

Algorithm 2 gives the elements of the Cholesky decomposition of a Hermitian Toeplitz matrix like the Lanczos-Phillips algorithm. Let T be a Hermitian Toeplitz matrix of order n,

$$T = [t_{i-j}]_{1 \le i,j \le n}. \tag{15}$$

Suppose all leading principal minors of T are nonzero. T has a unique decomposition $T = R^\top DR$ where R is unit upper triangular and D is diagonal. Note that $t_{-i} = \overline{t_i}$. Setting $s_{0,j} := t_{-j}$, we obtain the elements of DR: $(DR)_{i,j} = s_{i-1,j-1}$.

Let $F(x)$ be a power series such that

$$F(x) = t_0 + 2t_{-1}x + 2t_{-2}x^2 + \ldots. \tag{16}$$

Then $F(x)$ can be expressed by the Perron continued fraction [5]

$$F(x) = \cfrac{t_0}{1 - \cfrac{2u_0 x}{1 - u_0 x - \cfrac{v_1 x}{1 + u_1 x - \cfrac{v_2 x}{1 + u_2 x - \ddots}}}}.$$

The coefficients u_i and v_i have determinant expressions such that

$$u_i = \frac{T_{i-1,i-1} T_{i,-1}}{T_{i,i} T_{i-1,-1}}, \quad v_i = \frac{T_{i,-1}}{T_{i-1,-1}}. \tag{17}$$

Thus, Algorithm 2 can compute the coefficients of the Perron continued fraction. If a finite number of data $t_0, t_{-1}, \ldots, t_{-n}$ are given, $u_0, u_1, \ldots, u_n$ and $v_1, v_2, \ldots, v_n$ can be obtained by using the algorithm. In this case, we have a truncated continued fraction $\hat{F}(x)$. This is the Padé approximation of $F(x)$.

Algorithm 2 is useful to compute the Cholesky decomposition of a Hermitian Toeplitz matrix (15) and is essentially equivalent to the Schur algorithm [9] to solve the corresponding Toeplitz system $T\boldsymbol{x} = \boldsymbol{y}$. However, applying Algorithm 2 to a computation of a continued fraction through a determinant structure such as (13) and (17) has yet to be discussed. On the other hand, the discrete Schur flow can also compute the Perron continued fraction of (16) [5].

Similar to the previous section, let us consider the computation time and the percentage of failure between the two algorithms. For the computation time experiment, randomly generated $10^3, 5 \times 10^3$, and 10^4 moments in the range $[0, 1]$ are given. For the reliability experiment, randomly generated 10^3 moments are given, and we count the number of failures in 10^5 trials. Table 1 shows the environment conditions. Algorithm 2 is faster than the discrete Schur flow (Table 4), but both algorithms have similar reliabilities (Table 5).

Table 5. Failure percentage (%).

Algorithm 2	discrete Schur flow
0.0220	0.0230

5. Conclusion

We obtain a Hankel determinant expression of the general terms of the Lanczos-Phillips algorithm. Considering that the qd algorithm and the discrete Schur flow are discrete integrable systems with Hankel or Toeplitz determinant solutions, respectively, the Lanczos-Phillips algorithm (2) should also be a discrete integrable system. In discrete integrable systems, determinant solutions are assured to be positive by the theory of orthogonal polynomials. Therefore, the Lanczos-Phillips-type algorithms for both Hankel and Toeplitz matrices with nonzero leading principal minors should provide reliable computation methods for continued fractions that do not occur cancellation of significant digits and division by zero.

In the future, we plan to elucidate the relationship between the Lanczos-Phillips algorithm and discrete integrable systems as well as apply our algorithm method to other orthogonal polynomials.

References

[1] Y. Nakamura, K. Takasaki, S. Tsujimoto, M. Okado and J. Inoguchi, Kaisekigaku Hyakka II Kasekibunkei no Suuri (in Japanese), Asakura Shoten, Tokyo, 2018.

[2] J. L. Phillips, The triangular decomposition of Hankel matrices, Math. Comp., **25** (1971), 599–602.

[3] P. Henrici, Applied and Computational Complex Analysis. Vol. 2: Special Functions, Integral Transforms, Asymptotics, Continued Fractions, John Wiley & Sons, New York, 1977.

[4] H. Rutishauser, Der Quotienten-Differezen-Algorithmus, Z. Angew. Math. Phys., **5** (1954), 233–251.

[5] A. Mukaihira and Y. Nakamura, Schur flow for ortogonal polynomials on the unit circle and its integrable discretization, J. Comput. Appl. Math., **139** (2002), 75–94.

[6] A. S. Householder, The Theory of Matrices in Numerical Analysis, Pure and Appl. Sci., Blaisdell, Waltham, 1964.

[7] T. S. Chihara, An Introduction to Orthogonal Polynomials, Gordon and Breach Science Publishers, New York, 1978.

[8] G. Szegö, Orthogonal Polynomials, American Mathematical Society, Rhode Island, 1939.

[9] S.-Y. Kung and Y. H. Hu, A highly concurrent algorithm and pipelined architecture for solving Toeplitz systems, IEEE Trans. Acoust. Speech Signal Process., **31** (1983), 66–76.