Newton Sum Rules of Polynomials Defined by a Three-Term Recurrence Relation

P. NATALINI
Dipartimento di Matematica, Università degli Studi Roma III
Largo San Leonardo Murialdo, 1-00146 Roma, Italy
natalini@mat.uniroma3.it

P. E. RICCI
Dipartimento di Matematica, Università degli Studi di Roma "La Sapienza"
P.le A. Moro, 2-00185 Roma, Italy
ricci@uniroma1.it

(Received March 2000; revised and accepted January 2001)

Abstract—We derive a general formula for computing the Newton sum rules of every polynomial belonging to a given polynomial set. We use the following tools: a recursive computation of the coefficients of the given polynomial in terms of the coefficients of the three-term recurrence relation, the generalized Lucas polynomials of the first kind, and last, the Newton-Girard formulas. © 2001 Elsevier Science Ltd. All rights reserved.

Keywords—Generalized Lucas polynomials, Orthogonal polynomials, Jacobi matrix, Newton sum rules.

1. INTRODUCTION

In recent papers, many authors considered the problem of computing, in an efficient way, the Newton sum rules for the polynomials $P_N(x)$ belonging to a given orthogonal polynomial set \{${P_N(x)}_{N\in\mathbb{N}_0}$ (\mathbb{N}_0 := \mathbb{N} \cup \{0\}). We recall papers by Case [1], Buendia, Dehesa and Galvez [2], Ricci [3], Natalini [4], and Natalini and Ricci [5], where computation is performed in terms of the coefficients of a known differential equation, with polynomial coefficients, satisfied by the considered polynomials.

However, it is well known that there exist orthogonal polynomial sets for which it is not possible to construct such a differential equation or such an equation is not known. Therefore, it is important to derive a method by means of which the Newton sum rules can be constructed directly starting from the coefficients of the three-term recurrence relation. Some papers in this direction appeared in recent years such as Germano and Ricci [6] and Ifantis, Kokologiannaki and Siafarikas [7]. However, the procedure used there is quite cumbersome, since in the first one, the determinants of all principal minors of the Jacobi matrix appear, and in the second one, the powers of the same matrix are involved.

In this article, we derive a method of computation of the Newton sum rules simply constructing by recurrence the coefficients of the given polynomial in terms of the entries of the Jacobi matrix (i.e., the coefficients of the three-term recurrence relation). Then, we apply the generalized Lucas
polynomials of the first kind (see [8,9]) in order to compute recursively the Newton sum rules by using the Newton-Girard formulas.

The generalized Lucas polynomials of the first kind are defined as follows. Consider the bilateral linear homogeneous recurrence relation with \( N + 1 \) terms and constant coefficients \( u_{k,N} \), \( (k = 1, 2, \ldots, N; u_{N,N} \neq 0) \):

\[
X_m = u_{1,N}X_{m-1} - u_{2,N}X_{m-2} + \cdots + (-1)^{N-1}u_{N,N}X_{m-N}, \quad (m \in \mathbb{Z}),
\]

(1.1)

and the so-called characteristic equation

\[
x^N - u_{1,N}x^{N-1} + \cdots + (-1)^Nu_{N,N} = 0.
\]

(1.2)

In the simplest case, i.e., if equation (1.2) admits \( N \) distinct roots \( x_{1,N}, \ldots, x_{N,N} \), then the general solution of the recurrence relation (1.1) is expressed by

\[
X_m = C_1x_{1,N}^m + C_2x_{2,N}^m + \cdots + C_Nx_{N,N}^m, \quad (m \in \mathbb{Z}).
\]

(1.3)

A somewhat complicated formula holds when multiple roots appear. Among the solutions of equation (1.1), there is the sum of powers (for any fixed integral exponent) of the roots of the characteristic equation (1.2). This particular solution was called the primordial solution by Lucas [10], and will be introduced here in the following way.

For any integer \( N \), put

\[
\Psi_{N-1}(u_{1,N}, \ldots, u_{N,N}) = u_{1,N} = \sum_{k=1}^{N} x_{k,N},
\]

\[
\Psi_N(u_{1,N}, \ldots, u_{N,N}) = u_{1,N}^2 - 2u_{2,N} = \sum_{k=1}^{N} x_{k,N}^2,
\]

\[
\Psi_{N+1}(u_{1,N}, \ldots, u_{N,N}) = u_{1,N}^3 - 3u_{1,N}u_{2,N} + 3u_{3,N} = \sum_{k=1}^{N} x_{k,N}^3,
\]

(1.4)

\[
\Psi_{2N-2}(u_{1,N}, \ldots, u_{N,N}) = u_{1,N}\Psi_{2N-3} - u_{2,N}\Psi_{2N-4} + \cdots + (-1)^{N-2}u_{N-1,N}\Psi_{N-1} + (-1)^{N-1}u_{N,N} = \sum_{k=1}^{N} x_{k,N}^{2N-2},
\]

and for \( h > 2N - 2 \):

\[
\Psi_{h}(u_{1,N}, \ldots, u_{N,N}) = u_{1,N}\Psi_{h-1} - u_{2,N}\Psi_{h-2} + \cdots + (-1)^{h-N}u_{N-1,N}\Psi_{h-N+1} + (-1)^{N-1}u_{N,N}\Psi_{h-N+2} = \sum_{k=1}^{N} x_{k,N}^{h-N+2}.
\]

(1.5)

Then, according to the above definition, the generalized Lucas polynomial of the first kind \( \Psi_{h}(u_{1,N}, \ldots, u_{N,N}) \) gives the sum of the \( (h - N + 2) \)th powers of the roots of \( P_N(x) \), i.e., the Newton sum rule \( y_{h-N+2} \).

**Remark 1.1.** The choice of indices is justified in [9], in order to recover exactly, when \( N = 2 \), the classical Lucas polynomials of the first kind in two variables (see [10]).

In fact, if \( N = 2 \) and \( u_2 = 1 \), by putting \( u_1 = x \), we have

\[
\Psi_n(u_1, 1) = \Psi_n(x, 1) = 2T_n\left(\frac{x}{2}\right), \quad (n \in \mathbb{N}_0),
\]

where \( \{T_n(x)\}_{n\in\mathbb{N}_0} \) are the classical Chebyshev polynomials of the first kind.
If \( N \geq 3 \), and putting \( u_N = 1 \), we obtain a sequence of polynomials in \( N - 1 \) variables which appear as an extension of the classical Chebyshev polynomials of the first kind:

\[
\Psi_n(u_1, \ldots, u_{N-1}, 1) := T_n(u_1, \ldots, u_{N-1}), \quad (n \geq N - 2)
\]

(see [9]).

Further extensions of Chebyshev polynomials have been considered and studied by Lidl [11], Lidl et al. [12,13], Koornwinder [14,15], and Beerends [16].

2. RECURSIVE COMPUTATION OF THE COEFFICIENTS OF \( P_N(X) \)

Consider a polynomial set \( \{P_N(x)\}_{N \in \mathbb{N}_0} \) defined by the three-term recurrence relation

\[
\begin{align*}
P_{-1} &= 0, \\
P_N(x) &= (x - \alpha_N) P_{N-1}(x) - \beta_{N-1}^2 P_{N-2}(x), \quad (N \geq 1),
\end{align*}
\]

(2.1)

where the \( \alpha_N \) are real and \( \beta_{N-1} \neq 0 \). Then by a well-known Favard theorem (see [17, p. 21]), the polynomial \( P_N(x) \) belongs to an OPS with respect to a suitable measure (in general unknown).

We suppose all values \( \alpha_1, \alpha_2, \ldots, \alpha_N \) and \( \beta_0, \beta_1, \ldots, \beta_{N-1} \) are known (for every fixed \( N \)); put

\[
P_N(x) = \sum_{k=0}^{N} (-1)^k u_{k,N} x^{N-k}, \quad (N = 1, 2, \ldots),
\]

(2.2)

and denote by \( x_{1,N}, \ldots, x_{N,N} \), the zeros of \( P_N(x) \), and by

\[
y_h = \sum_{k=1}^{N} x_h^{k,N}, \quad (h = 1, 2, \ldots),
\]

(2.3)

the corresponding Newton sum rules. Then we can state the following theorem.

**Theorem 2.1.** For the coefficients (up to a change of sign) \( u_{k,N} \) of the fixed polynomial \( P_N(x) \), the recurrence relation holds true:

\[
u_{k,N} = u_{k,N-1} + \alpha_N u_{k-1,N-1} - \beta_{N-1}^2 u_{k-2,N-2},
\]

(2.4)

where \( N = 1, 2, \ldots \) and \( k = 1, \ldots, N \). In the above formula, the coefficients \( u_{N,N-1}, u_{-1,N-1}, u_{-1,N-2}, u_{-2,N-2} \) are assumed to be zero.

**Proof.** The proof is simply obtained by substituting (2.2) corresponding to values \( N, N-1, N-2 \) into the recurrence relation (2.1), and then equating the coefficients of corresponding powers of \( x \).

The above considerations (i.e., Theorem 2.1 and definitions (1.4),(1.5)) can be formalized in the following result.

**Theorem 2.2.** Consider a polynomial \( P_N(x) \), given by (2.2), which satisfies recurrence relation (2.1). Then the coefficients of \( P_N(x) \) are recursively linked to the entries of the Jacobi matrix \( \alpha_1, \alpha_2, \ldots, \alpha_n \) and \( \beta_0, \beta_1, \ldots, \beta_{n-1} \) by formula (2.4), and for the Newton sum rules, the following representation formula holds true:

\[
y_h = \sum_{k=1}^{N} x_h^{k,N} = \Psi_{h+N-2}(u_{1,N}, \ldots, u_{N,N}).
\]

(2.5)

**Proof.** Substituting \( h \) with \( h + N - 2 \) into equation (1.5), equation (2.5) immediately comes out.
This formula, provided that initial conditions (1.4) are computed, permits recursive computation of moments via (1.5).

Note that a recursive computation of the Newton sum rules is possible starting by the preceding iterative formula for the Lucas polynomials of the first kind using, as a starting set of values, the so-called Newton-Girard formulas:

\[
\begin{align*}
    u_{1,N} &= y_1, \\
    u_{2,N} &= \frac{1}{2} (u_{1,N}y_1 - y_2) = \frac{1}{2} (y_1^2 - y_2), \\
    u_{3,N} &= \frac{1}{3} (-u_{1,N}y_2 + u_{2,N}y_1 + y_3) = \frac{1}{6} (y_1^3 - 3y_1y_2 + 2y_3), \\
    \vdots \\
    u_{N,N} &= \frac{1}{N} \left\{ \left[ \left( \sum_{j=1}^{N-1} u_{j,N}y_{N-j} \right) - \prod_{j=1}^{N-1} y_j \right] + (-1)^{N-1} y_N \right\}.
\end{align*}
\]

The algorithm works as follows.

1. Compute the coefficients \( u_{1,N}, u_{2,N}, \ldots, u_{N,N} \) by using the recurrence relation (2.4) in terms of the entries of the Jacobi matrix.
2. Compute the first \( N \) Newton sum rules inverting the Newton-Girard formulas taking into account the linearity of the \( h \)th equation with respect to \( y_h \) (\( h = 1, 2, \ldots, N \)).
3. Use the coefficients \( u_{1,N}, u_{2,N}, \ldots, u_{N,N} \), and computed values of the \( y_r \) (\( r = 1, 2, \ldots, N \)) in order to start the recurrence relation (1.5).

This algorithm gives all the subsequent values for the Newton sum rules in a very efficient and fast way.

### 3. NUMERICAL RESULTS

By using the above-mentioned algorithm, we have computed numerically the first moments of the so-called scaled co-recursive associated Jacobi polynomials \( \{ P_{n}^{(a,b)}(x; \beta, \gamma, c) \} \) which satisfy a three-term recurrence relation of type (2.1), with

\[
\begin{align*}
    \alpha &= \frac{b^2 - a^2}{(2N + 2c + a + b - 2)(2N + 2c + a + b)[1 + (\gamma - 1)\delta_{N-1,0}]} + \frac{\beta\delta_{N-1,0}}{1 + (\gamma - 1)\delta_{N-1,0}}, \\
    \beta &= \frac{4(N + c - 1)(N + c + a - 1)(N + c + b - 1)}{(2N + 2c + a + b - 2)^2(2N + 2c + a + b - 1)(2N + 2c + a + b - 3)[1 + (\gamma - 1)\delta_{N-1,0}][1 + (\gamma - 1)\delta_{N-2,0}]},
\end{align*}
\]

where \( \delta_{i,j} \) is the Dirac delta. We have used a Fortran77 program, written by the first author, and a simple personal computer. The execution time of the program is extremely fast. The obtained results are only affected by the error machine.

| \( m_1 \) | \( m_{12} \) | \( m_{13} \) | \( m_{14} \) | \( m_{15} \) | \( m_{16} \) | \( m_{17} \) | \( m_{18} \) | \( m_{19} \) | \( m_{20} \) | \( m_{21} \) | \( m_{22} \) | \( m_{23} \) | \( m_{24} \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.12433394224979e + 00 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 | 0.14268877994674e + 04 |

Table 1.
In Table 1, we have considered the parameters $N = 15$, $a = 1$, $b = 1/2$, $c = 1$, $\beta = 1$, and $\gamma = 1/2$.

In Table 2, we present the case of parameters $N = 45$, $a = 2$, $b = 1$, $c = 2$, $\beta = 1$, and $\gamma = 1/2$.

REFERENCES