

MULTISTEP ε -ALGORITHM, SHANKS' TRANSFORMATION,
AND THE LOTKA-VOLTERRA SYSTEM
BY HIROTA'S METHOD

CLAUDE BREZINSKI, YI HE, XING-BIAO HU, MICHELA REDIVO-ZAGLIA,
AND JIAN-QING SUN

ABSTRACT. In this paper, we propose a multistep extension of the Shanks sequence transformation. It is defined as a ratio of determinants. Then, we show that this transformation can be recursively implemented by a multistep extension of the ε -algorithm of Wynn. Some of their properties are specified. Thereafter, the multistep ε -algorithm and the multistep Shanks transformation are proved to be related to an extended discrete Lotka–Volterra system. These results are obtained by using Hirota's bilinear method, a procedure quite useful in the solution of nonlinear partial differential and difference equations.

1. THE SCENERY

Let (S_n) be a sequence of numbers converging to S . If its convergence is slow, it can be transformed, by a *sequence transformation*, into a set of new sequences $\{(T_k^{(n)})\}$, depending on two indexes k and n , and converging, under certain assumptions, faster to the same limit, that is such that

$$\lim_{n \rightarrow \infty} \frac{T_k^{(n)} - S}{S_n - S} = 0, \quad \text{or} \quad \lim_{k \rightarrow \infty} \frac{T_k^{(n)} - S}{S_k - S} = 0, \quad \text{or both.}$$

A well-known example of such a transformation is the Richardson extrapolation process, which gives rise to the Romberg's method for accelerating the convergence of the trapezoidal rule for approximating a definite integral. Let us mention that sequence transformations can also be applied to diverging power sequences, thus leading, in some situations, to interesting results such as analytic continuation (this is the case of the ε -algorithm which, applied to the partial sum of a divergent power series, computes its Padé approximants).

In many sequence transformations, the terms of the new sequences can be expressed as ratios of determinants, and there exists, in each particular case, a (usually nonlinear) *recursive algorithm* for avoiding the computation of these determinants and implementing the transformation under consideration [11, 39, 44, 43, 45].

The most well-known transformation of this type is due to Shanks [37, 38]. It can be implemented via the ε -algorithm of Wynn [46]. Recently, a new recursive algorithm for accelerating the convergence of sequences was derived by He, Hu, Sun

Received by the editor December 21, 2010 and, in revised form, March 14, 2011.

2010 *Mathematics Subject Classification*. Primary 65B05, 39A14, 37K10.

Key words and phrases. Convergence acceleration algorithm, ε -algorithm, Shanks' transformation, Lotka–Volterra system.

and Weniger [15] from the lattice Boussinesq equation. This algorithm resembles the ε -algorithm, and it was proved that the quantities it computes can be expressed as ratios of determinants, thus extending the Shanks sequence transformation. In this paper, inspired by this approach, we will extend further the Shanks transformation, and we will show that it can be implemented by an extension of the ε -algorithm, thus leading to a multistep Shanks transformation and a multistep ε -algorithm. The proof makes use of Hirota's bilinear method [18] which was invented for resolving integrable nonlinear partial differential or difference evolution equations having soliton solutions.

For some years now, there has been a great concern for convergence acceleration algorithms among the community of mathematical physicists working on integrable systems, KdV and other equations, soliton theory, Toda lattices, etc. [9, 25, 26, 32, 33]. These researchers are interested by the fact that convergence acceleration algorithms are nonlinear difference equations in two variables whose solutions are explicitly known. Determinants often play a central role in this type of problem as exemplified, for example, in [42]. An important procedure for obtaining a closed-form solution of soliton equations is Hirota's bilinear method [18] which consists of writing the solution as a ratio, and then working with its numerator and its denominator.

In Section 2, we recall the Shanks sequence transformation and how it can be implemented by the ε -algorithm of Wynn. In Section 3, we present our multistep extension of the Shanks transformation, and the corresponding multistep extension of the ε -algorithm. The quantities involved in this transformation and in this algorithm are expressed by ratios of determinants generalizing those of Hankel. Section 4 is devoted to some relations between these determinants. Hirota's bilinear method is presented in Section 5, and it is used for deriving relations between the numerators and the denominators of the quantities computed by the multistep ε -algorithm. In Section 6, we show that the quantities recursively computed by this algorithm correspond to the ratios of determinants defining the multistep Shanks transformation. We will also verify that the multistep Shanks transformation and the multistep ε -algorithm produce identical results. Section 7 will be devoted to some properties of the multistep Shanks transformation. Finally, in Section 8, the connection between an extended discrete hungry Lotka–Volterra system and the multistep ε -algorithm is discussed. Hirota's bilinear method is essential for obtaining these results. The paper ends with some considerations for further research.

2. THE SHANKS TRANSFORMATION AND THE ε -ALGORITHM

The Shanks sequence transformation [37, 38] $e_k : (S_n) \mapsto \{(e_k(S_n))\}$ consists of transforming a given sequence (S_n) into the set of sequences $\{(e_k(S_n))\}$ whose terms are defined by

$$(2.1) \quad e_k(S_n) = \frac{\mathcal{H}_{k+1}(S_n)}{\mathcal{H}_k(\Delta^2 S_n)}, \quad k, n = 0, 1, \dots,$$

where Δ is the usual forward difference operator whose powers are defined by

$$\Delta^{i+1} S_n = \Delta^i S_{n+1} - \Delta^i S_n$$

with $\Delta^0 S_n = S_n$, and where $\mathcal{H}_k(u_n)$ denotes the Hankel determinant

$$\mathcal{H}_k(u_n) = \begin{vmatrix} u_n & u_{n+1} & \cdots & u_{n+k-1} \\ u_{n+1} & u_{n+2} & \cdots & u_{n+k} \\ \vdots & \vdots & \cdots & \vdots \\ u_{n+k-1} & u_{n+k} & \cdots & u_{n+2k-2} \end{vmatrix},$$

with $\mathcal{H}_0(u_n) = 1$.

Obviously, replacing each row, in this determinant, by its difference with the previous one, repeating this operation several times, and performing it also on the columns, we have

$$\mathcal{H}_k(u_n) = \begin{vmatrix} u_n & \cdots & u_{n+k-1} \\ \Delta u_n & \cdots & \Delta u_{n+k-1} \\ \Delta^2 u_n & \cdots & \Delta^2 u_{n+k-1} \\ \vdots & \cdots & \vdots \\ \Delta^{k-1} u_n & \cdots & \Delta^{k-1} u_{n+k-1} \end{vmatrix} = \begin{vmatrix} u_n & \Delta u_n & \cdots & \Delta^{k-1} u_n \\ \Delta u_n & \Delta^2 u_n & \cdots & \Delta^k u_n \\ \vdots & \vdots & \cdots & \vdots \\ \Delta^{k-1} u_n & \Delta^k u_n & \cdots & \Delta^{2k-2} u_n \end{vmatrix}.$$

The ε -algorithm is a recursive algorithm due to Wynn [46] for implementing the Shanks transformation without computing the Hankel determinants appearing in (2.1). Its rule is

$$(2.2) \quad \varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}}, \quad k, n = 0, 1, \dots$$

with $\varepsilon_{-1}^{(n)} = 0$ and $\varepsilon_0^{(n)} = S_n$, for $n = 0, 1, \dots$

The connection between the ε -algorithm and the Shanks transformation is given by

$$(2.3) \quad \varepsilon_{2k}^{(n)} = e_k(S_n) \quad \text{and} \quad \varepsilon_{2k+1}^{(n)} = \frac{1}{e_k(\Delta S_n)}, \quad k, n = 0, 1, \dots$$

Thus, the $\varepsilon_{2k+1}^{(n)}$'s are intermediate results, and we have

$$\varepsilon_{2k}^{(n)} = \frac{\mathcal{H}_{k+1}(S_n)}{\mathcal{H}_k(\Delta^2 S_n)} \quad \text{and} \quad \varepsilon_{2k+1}^{(n)} = \frac{\mathcal{H}_k(\Delta^3 S_n)}{\mathcal{H}_{k+1}(\Delta S_n)}.$$

The quantities $\varepsilon_k^{(n)}$ are usually displayed in a two-dimensional array (the ε -array) where the lower index k remains the same in a column of the table, and the upper index n is the same in a descending diagonal. Thus, the rule (2.2) relates four quantities located at the four vertices of a lozenge in three different columns and two descending diagonals as showed below

$$\begin{array}{ccc} & \varepsilon_k^{(n)} & \\ \varepsilon_{k-1}^{(n+1)} & & \varepsilon_{k+1}^{(n)} \\ & \varepsilon_k^{(n+1)} & \end{array}$$

For implementing the ε -algorithm efficiently, the best technique, due to Wynn [48, 49], consists of storing the last ascending diagonal of the ε -array (in this diagonal the sum of the lower and the upper indexes is constant), and to add, one by one, the terms of the sequence to be transformed. Then, a new ascending diagonal is built step-by-step, by moving up the lozenge, and the new diagonal gradually replaces the old one. The corresponding FORTRAN subroutine can be found in [11].

Since the quantities with an odd lower index are intermediate computations, they can be eliminated, thus leading to the *cross rule* also due to Wynn [50]

$$\frac{1}{\varepsilon_{2k+4}^{(n)} - \varepsilon_{2k+2}^{(n+1)}} + \frac{1}{\varepsilon_{2k}^{(n+2)} - \varepsilon_{2k+2}^{(n+1)}} = \frac{1}{\varepsilon_{2k+2}^{(n+2)} - \varepsilon_{2k+2}^{(n+1)}} + \frac{1}{\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k+2}^{(n+1)}},$$

with the initial conditions $\varepsilon_{-2}^{(n)} = \infty$ and $\varepsilon_0^{(n)} = S_n$ for $n = 0, 1, \dots$. Obviously, it is also possible to eliminate the $\varepsilon_k^{(n)}$'s with an even lower index for obtaining a rule only involving quantities with a lower odd index, although this is less useful from the numerical point of view.

The proof given by Wynn for his ε -algorithm was mostly a *verification* of the link between the Shanks transformation and the algorithm, since he introduced the ratios of Hankel determinants for $e_k(S_n)$ and $e_k(\Delta S_n)$ into the rule of the ε -algorithm, and he showed that the equality held by making use of Sylvester's determinantal identity and Schweins' determinantal identity which can be found, for example, in [1] (see [8, pp. 142–143] for their proofs). The difficulty of the proof resided in the nonlinearity of the algorithm. Of course, Wynn's great merit was the idea of the ε -algorithm itself, followed by this verification.

There are three approaches for linking a sequence transformation and a (usually nonlinear) recursive algorithm for its implementation. By increasing order of complexity, they are:

- (1) *Verification*: the transformation and the algorithm are both known, and one has to *verify* that they lead to identical sequences. This is the method followed by Wynn in [46] when he gave his ε -algorithm.
- (2) *Derivation*: only the transformation is known, and one has to *derive* an algorithm for its implementation. This is the case, for example, of the E -transformation which is the most general sequence transformation known so far, and which can be implemented by the E -algorithm, an algorithm which appeared almost simultaneously in various contexts [6, 14, 23, 36]. This was also certainly the way Wynn followed when he derived his ε -algorithm, although it was not presented like that in his paper [46].
- (3) *Proof*: only the algorithm is known, and one has to guess a formula (that is a ratio of determinants) for the transformation it is implementing, and to *prove* it. This was the situation for the second generalization of the ε -algorithm proposed in [5], whose form was obtained by Salam [34, 35]. Let us mention that the θ -algorithm [4] is an extrapolation algorithm for which no determinantal formula is yet known, if it exists.

Now, after presenting the multistep Shanks transformation and the multistep ε -algorithm (Section 3), we will show, with the help of determinantal identities (Section 4) and Hirota's bilinear method (Section 5), that the multistep ε -algorithm is a possible algorithm for implementing the multistep Shanks transformation.

3. THE MULTISTEP SHANKS TRANSFORMATION AND THE MULTISTEP ε -ALGORITHM

Let us define the *multistep Shanks transformation* $e_{k,m} : (S_n) \mapsto \{(e_{k,m}(S_n))\}$, where $m \geq 1$ is a fixed integer, by

$$(3.1) \quad e_{k,m}(S_n) = \frac{H_{k+1}(S_n)}{H_k(\Delta^{m+1} S_n)}, \quad k, n = 0, 1, \dots,$$

and, thus, that

$$(3.5) \quad e_{k,m}(S_n) = \varepsilon_{(m+1)k,m}^{(n)}.$$

This relation shows that only the quantities $\varepsilon_{k,m}^{(n)}$ whose first lower index is a multiple of $m + 1$ are interesting for the purpose of convergence acceleration. All the other ones are intermediate, but needed, quantities. The computation of $e_{k,m}(S_n) = \varepsilon_{(m+1)k,m}^{(n)}$ needs the knowledge of $S_n, \dots, S_{n+(m+1)k}$.

For the intermediate computations, in the same theorem, we will see that for all k and n , it also holds that

$$(3.6) \quad \varepsilon_{(m+1)(k-1)+1,m}^{(n)} = \frac{H_{k-1}(\Delta^{m+2}S_n)}{H_k(\Delta S_n)},$$

$$(3.7) \quad \varepsilon_{(m+1)(k-1)+i,m}^{(n)} = \frac{\Phi_{k+1}(\Delta^{i-1}S_n)}{H_k(\Delta^i S_n)}, \quad i = 2, 3, \dots, m,$$

where the determinants H_k are defined as above, and the determinants Φ_k , which also depend on m (dependence not indicated again), are given by

$$\Phi_k(u_n) = \begin{vmatrix} n & n+1 & \cdots & n+k-1 \\ u_n & u_{n+1} & \cdots & u_{n+k-1} \\ \Delta^m u_n & \Delta^m u_{n+1} & \cdots & \Delta^m u_{n+k-1} \\ \Delta^{2m} u_n & \Delta^{2m} u_{n+1} & \cdots & \Delta^{2m} u_{n+k-1} \\ \vdots & \vdots & \cdots & \vdots \\ \Delta^{(k-2)m} u_n & \Delta^{(k-2)m} u_{n+1} & \cdots & \Delta^{(k-2)m} u_{n+k-1} \end{vmatrix},$$

$k = 1, 2, \dots, \quad n = 0, 1, \dots,$

with $\Phi_{-1}(u_n) = 0$ and $\Phi_0(u_n) = 1$.

Let us remark that, from (3.6), we have

$$\varepsilon_{(m+1)k+1,m}^{(n)} = \frac{1}{e_{k,m}(\Delta S_n)}, \quad k, n = 0, 1, \dots,$$

a result similar to the second relation (2.3) for the ε -algorithm of Wynn, that it is recovered when $m = 1$.

In the next sections, we will link the multistep Shanks transformation (3.1) and the multistep ε -algorithm (3.2). For doing this, we will follow a procedure similar, although more difficult, to the procedure used in [15] (which is also based on Hirota's bilinear method) for deriving a determinantal expression for a new acceleration algorithm obtained from the lattice Boussinesq equation.

Let us explain, in more detail, the chaining of the successive steps of our approach. First, in Section 4, some relations between the determinants $H_k(\Delta^i S_n)$ and $\Phi_k(\Delta^i S_n)$ will be established. We will only employ Sylvester's determinantal identity (which is, in fact, Jacobi's determinantal identity after a permutation of rows and columns), contrarily to the proofs given in [46] and [15] where Schweins' determinantal identity is also used. Then, in Section 5, Hirota's bilinear method will be applied to the multistep ε -algorithm (3.2). In Section 6, by using the results of Section 5, we will show that the quantities computed by the multistep ε -algorithm (3.2) can be expressed by the determinantal relation (3.4) (interesting quantities), and (3.6)–(3.7) (intermediate computations). Then, thanks to (3.5), the multistep ε -algorithm is one of the algorithms for implementing the multistep Shanks transformation (3.1). Moreover, in the same section, starting from the

relations (3.4), (3.6) and (3.7), we will verify that the recursive rule (3.2) of the multistep ε -algorithm produces these quantities.

4. RELATIONS BETWEEN DETERMINANTS

Let A be a square matrix, α, β, γ and δ numbers, a, b, c and d vectors of the same dimension as A . Let M be the matrix

$$M = \begin{pmatrix} \alpha & a^T & \beta \\ b & A & c \\ \gamma & d^T & \delta \end{pmatrix}.$$

Sylvester's determinantal identity is

$$|M| \cdot |A| = \begin{vmatrix} \alpha & a^T \\ b & A \end{vmatrix} \cdot \begin{vmatrix} A & c \\ d^T & \delta \end{vmatrix} - \begin{vmatrix} a^T & \beta \\ A & c \end{vmatrix} \cdot \begin{vmatrix} b & A \\ \gamma & d^T \end{vmatrix}.$$

Let us now prove some determinantal identities between the determinants $H_k(\Delta^i S_n)$ and $\Phi_k(\Delta^i S_n)$, defined in Section 3, that will be useful in the sequel.

Lemma 4.1.

$$(4.1) \quad \begin{aligned} H_{k+1}(\Delta S_n)H_k(\Delta^m S_{n+1}) = \\ H_k(\Delta^{m+1} S_n)H_{k+1}(S_{n+1}) - H_k(\Delta^{m+1} S_{n+1})H_{k+1}(S_n). \end{aligned}$$

Proof. We consider the determinant

$$D_1 = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta^{i+m} S_n & \Delta^{i+m} S_{n+1} & \cdots & \Delta^{i+m} S_{n+k+1} \\ \vdots & \vdots & & \vdots \\ \Delta^{i+km} S_n & \Delta^{i+km} S_{n+1} & \cdots & \Delta^{i+km} S_{n+k+1} \\ \Delta^i S_n & \Delta^i S_{n+1} & \cdots & \Delta^i S_{n+k+1} \end{vmatrix} = (-1)^k H_{k+1}(\Delta^{i+1} S_n).$$

The second expression for D_1 is obtained by replacing each column, from the last one, by its difference with the previous one. Thus, we get a determinant whose first row only contains 0 except in the first column where the element is equal to 1. Expanding this determinant with respect to its first row, and putting its last row as the first one, we see that $D_1 = (-1)^k H_{k+1}(\Delta^{i+1} S_n)$. Let us now apply Sylvester's identity to the first expression of D_1 , and perform a similar manipulation on the rows and the columns of the other determinants, we obtain

$$(4.2) \quad \begin{aligned} H_{k+1}(\Delta^{i+1} S_n)H_k(\Delta^{i+m} S_{n+1}) = \\ H_k(\Delta^{i+m+1} S_n)H_{k+1}(\Delta^i S_{n+1}) - H_k(\Delta^{i+m+1} S_{n+1})H_{k+1}(\Delta^i S_n). \end{aligned}$$

Setting $i = 0$ in this relation, we get (4.1). □

A similar identity, which will be used in the sequel, also holds if S_n is replaced by ΔS_n , that is for $i = 1$.

Lemma 4.2.

$$(4.3) \quad \begin{aligned} H_k(\Delta^{i+1} S_n)H_{k-1}(\Delta^i S_{n+1}) = \\ H_{k-1}(\Delta^{i+1} S_n)H_k(\Delta^i S_{n+1}) - H_{k-1}(\Delta^{i+1} S_{n+1})H_k(\Delta^i S_n). \end{aligned}$$

Proof. Let D_2 be the determinant obtained from D_1 by replacing k by $k - 1$, and moving the last row to the second position. Replacing each column, from the last one, by its difference with the previous one, we see that $D_2 = H_k(\Delta^{i+1} S_n)$, and, applying Sylvester's identity to it, we get (4.3). □

Lemma 4.3.

$$(4.4) \quad \begin{aligned} H_k(\Delta S_n)H_k(\Delta^m S_{n+1}) = \\ H_k(\Delta^{m+1} S_n)H_k(S_{n+1}) - H_{k+1}(S_n)H_{k-1}(\Delta^{m+1} S_{n+1}). \end{aligned}$$

Proof. Setting $i = m$ in (4.3), we have

$$(4.5) \quad \begin{aligned} H_k(\Delta^{m+1} S_n)H_{k-1}(\Delta^m S_{n+1}) = \\ H_{k-1}(\Delta^{m+1} S_n)H_k(\Delta^m S_{n+1}) - H_{k-1}(\Delta^{m+1} S_{n+1})H_k(\Delta^m S_n). \end{aligned}$$

Applying now Sylvester’s identity to the determinant $H_{k+1}(\Delta^i S_n)$, we get

$$\begin{aligned} H_{k+1}(\Delta^i S_n)H_{k-1}(\Delta^{i+m} S_{n+1}) = \\ H_k(\Delta^i S_n)H_k(\Delta^{i+m} S_{n+1}) - H_k(\Delta^i S_{n+1})H_k(\Delta^{i+m} S_n). \end{aligned}$$

Setting $i = 0$ in this relation, we obtain

$$(4.6) \quad H_{k+1}(S_n)H_{k-1}(\Delta^m S_{n+1}) = H_k(S_n)H_k(\Delta^m S_{n+1}) - H_k(S_{n+1})H_k(\Delta^m S_n).$$

Then, we multiply (4.5) by $H_k(S_{n+1})$, we multiply (4.6) by $H_{k-1}(\Delta^{m+1} S_{n+1})$, we subtract, and get

$$\begin{aligned} H_{k-1}(\Delta^m S_{n+1})[H_k(\Delta^{m+1} S_n)H_k(S_{n+1}) - H_{k+1}(S_n)H_{k-1}(\Delta^{m+1} S_{n+1})] = \\ H_k(\Delta^m S_{n+1})[H_{k-1}(\Delta^{m+1} S_n)H_k(S_{n+1}) - H_k(S_n)H_{k-1}(\Delta^{m+1} S_{n+1})]. \end{aligned}$$

Using (4.1), we see that the bracket in the right-hand side is equal to $H_k(\Delta S_n)H_{k-1}(\Delta^m S_{n+1})$. After simplifying both sides by $H_{k-1}(\Delta^m S_{n+1})$, we obtain (4.4). \square

A similar identity, which will be used in the sequel, also holds if S_n is replaced by ΔS_n .

Lemma 4.4.

$$(4.7) \quad \begin{aligned} H_k(\Delta^i S_{n+1})H_{k-1}(\Delta^{i+2} S_n) = \\ H_k(\Delta^{i+1} S_n)\Phi_k(\Delta^i S_{n+1}) - H_{k-1}(\Delta^{i+1} S_{n+1})\Phi_{k+1}(\Delta^i S_n). \end{aligned}$$

Proof. We consider the determinant

$$D_3 = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta^i S_n & \Delta^i S_{n+1} & \cdots & \Delta^i S_{n+k+1} \\ \Delta^{i+m} S_n & \Delta^{i+m} S_{n+1} & \cdots & \Delta^{i+m} S_{n+k+1} \\ \vdots & \vdots & \cdots & \vdots \\ \Delta^{i+(k-1)m} S_n & \Delta^{i+(k-1)m} S_{n+1} & \cdots & \Delta^{i+(k-1)m} S_{n+k+1} \\ n & n+1 & \cdots & n+k+1 \end{vmatrix}.$$

Obviously, we also have

$$\begin{aligned}
 D_3 &= (-1)^k \begin{vmatrix} 1 & 1 & \cdots & 1 \\ n & n+1 & \cdots & n+k+1 \\ \Delta^i S_n & \Delta^i S_{n+1} & \cdots & \Delta^i S_{n+k+1} \\ \vdots & \vdots & & \vdots \\ \Delta^{i+(k-1)m} S_n & \Delta^{i+(k-1)m} S_{n+1} & \cdots & \Delta^{i+(k-1)m} S_{n+k+1} \end{vmatrix} \\
 &= (-1)^k \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta^{i+1} S_n & \Delta^{i+1} S_{n+1} & \cdots & \Delta^{i+1} S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta^{i+1+(k-1)m} S_n & \Delta^{i+1+(k-1)m} S_{n+1} & \cdots & \Delta^{i+1+(k-1)m} S_{n+k} \end{vmatrix} \\
 &= (-1)^k H_k(\Delta^{i+2} S_n).
 \end{aligned}$$

Now we apply Sylvester's identity to the first expression of D_3 given above, and replace D_3 by $(-1)^k H_k(\Delta^{i+2} S_n)$. We get, after similar manipulations on the columns of the other determinants,

$$\begin{aligned}
 (4.8) \quad &H_k(\Delta^{i+2} S_n)H_k(\Delta^i S_{n+1}) = \\
 &H_k(\Delta^{i+1} S_n)\Phi_{k+1}(\Delta^i S_{n+1}) - H_k(\Delta^{i+1} S_{n+1})\Phi_{k+1}(\Delta^i S_n).
 \end{aligned}$$

Then, we apply the Sylvester's identity to the determinant $\Phi_{k+1}(\Delta^i S_n)$. We get

$$\Phi_{k+1}(\Delta^i S_n)H_{k-1}(\Delta^i S_{n+1}) = \Phi_k(\Delta^i S_n)H_k(\Delta^i S_{n+1}) - \Phi_k(\Delta^i S_{n+1})H_k(\Delta^i S_n).$$

We multiply this identity by $H_{k-1}(\Delta^{i+1} S_{n+1})$, we multiply (4.3) by $\Phi_k(\Delta^i S_{n+1})$, we subtract, and get

$$\begin{aligned}
 &H_{k-1}(\Delta^i S_{n+1})[\Phi_{k+1}(\Delta^i S_n)H_{k-1}(\Delta^{i+1} S_{n+1}) - H_k(\Delta^{i+1} S_n)\Phi_k(\Delta^i S_{n+1})] = \\
 &H_k(\Delta^i S_{n+1})[\Phi_k(\Delta^i S_n)H_{k-1}(\Delta^{i+1} S_{n+1}) - H_{k-1}(\Delta^{i+1} S_n)\Phi_k(\Delta^i S_{n+1})].
 \end{aligned}$$

Using (4.8), we see that the bracket in the right-hand side is equal to $-H_{k-1}(\Delta^{i+2} S_n)H_{k-1}(\Delta^i S_{n+1})$. After simplifying both sides by $H_{k-1}(\Delta^i S_{n+1})$, we obtain (4.7). \square

Lemma 4.5.

$$\begin{aligned}
 (4.9) \quad &H_k(\Delta S_n)H_{k-2}(\Delta^{m+1} S_{n+1}) = \\
 &H_{k-1}(\Delta^{m+1} S_{n+1})H_{k-1}(\Delta S_n) - H_{k-1}(\Delta^{m+1} S_n)H_{k-1}(\Delta S_{n+1}).
 \end{aligned}$$

Proof. We consider the determinant

$$D_4 = \begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+k} \\ 1 & 1 & \cdots & 1 \\ \Delta^m S_n & \Delta^m S_{n+1} & \cdots & \Delta^m S_{n+k} \\ \vdots & \vdots & & \vdots \\ \Delta^{(k-1)m} S_n & \Delta^{(k-1)m} S_{n+1} & \cdots & \Delta^{(k-1)m} S_{n+k} \end{vmatrix}.$$

After exchanging the first row and the second row, we see that $D_4 = -H_k(\Delta S_n)$. Let us now apply Sylvester's identity to D_4 , and perform a similar manipulation on the first and second rows of the various determinants. We obtain (4.9). \square

5. HIROTA’S BILINEAR METHOD

Hirota’s bilinear method [18] is a technique which could be much useful for solving certain nonlinear differential and difference equations. It consists of expressing the unknown as a ratio and, then, in treating separately the numerator and the denominator.

We will now apply this method to the multistep ε -algorithm, and set

$$(5.1) \quad \varepsilon_{k,m}^{(n)} = \frac{G_{k,m}^n}{F_{k,m}^n}.$$

The idea is to consider the recursive relation (3.2) and to try to find the determinants $G_{k,m}^n$ and $F_{k,m}^n$ for which the equality holds. Let us notice that the numerator and the denominator in (5.1) are not uniquely defined, but only up to a common multiplying factor.

As we will see, the determinants $F_{k,m}^n$ will be given by the same formula for all values of the index k . On the contrary, there will be three different expressions for the determinants $G_{k,m}^n$, depending on the fact that the first subscript of $\varepsilon_{k,m}^{(n)}$ belongs to a column that is a multiple of $m + 1$ (denoted by $(m + 1)k$), or belongs to the next column (denoted by $(m + 1)k + 1$), or, finally, belongs to the successive columns up to the column that is the next multiple of $m + 1$ (denoted by $(m + 1)k + j$, for $j = 2, \dots, m$) exclusively.

For simplicity, in the sequel, we will omit the subscript that indicates the dependence on the fixed integer m in $G_{k,m}^n$ and in $F_{k,m}^n$, since it is fixed.

We first have the following lemma that states, by using (5.1), a possible (but not unique) identity for the recursive rule (3.2) of the multistep ε -algorithm.

Lemma 5.1.

$$(5.2) \quad (F_{k+m+1}^n G_k^{n+1} - F_k^{n+1} G_{k+m+1}^n) \prod_{i=1}^m (F_{k+i}^n G_{k+i}^{n+1} - F_{k+i}^{n+1} G_{k+i}^n) = - F_{k+1}^n F_{k+m}^{n+1} \prod_{i=1}^m F_{k+i+1}^n F_{k+i-1}^{n+1}.$$

Proof. Plugging (5.1) into the recursive rule (3.2) of the ε -algorithm, we get

$$(5.3) \quad \frac{G_{k+1}^n}{F_{k+1}^n} - \frac{G_{k-m}^{n+1}}{F_{k-m}^{n+1}} = \frac{1}{\prod_{i=1}^m \left(\frac{G_{k-m+i}^{n+1}}{F_{k-m+i}^{n+1}} - \frac{G_{k-m+i}^n}{F_{k-m+i}^n} \right)}$$

$$\frac{F_{k-m}^{n+1} G_{k+1}^n - F_{k+1}^n G_{k-m}^{n+1}}{F_{k+1}^n F_{k-m}^{n+1}} = \frac{\prod_{i=1}^m F_{k-m+i}^n F_{k-m+i}^{n+1}}{\prod_{i=1}^m (F_{k-m+i}^n G_{k-m+i}^{n+1} - F_{k-m+i}^{n+1} G_{k-m+i}^n)}.$$

Now, we cross-multiply the numerator of one side by the denominator of the other side, and we equate both sides. Replacing k by $k + m$ and changing the sign, the equation (5.3) becomes (5.2) since

$$F_{k+m+1}^n F_k^{n+1} \prod_{i=1}^m F_{k+i}^n F_{k+i}^{n+1} = \prod_{i=1}^{m+1} F_{k+i}^n \prod_{i=0}^m F_{k+i}^{n+1}. \quad \square$$

The second preliminary result is contained in the following.

Lemma 5.2. *If the following relations hold,*

$$(5.4) \quad F_{(m+1)k+1}^n G_{(m+1)k+1}^{n+1} - F_{(m+1)k+1}^{n+1} G_{(m+1)k+1}^n = -F_{(m+1)k+2}^n F_{(m+1)k}^{n+1},$$

$$(5.5) \quad F_{(m+1)k+1}^n G_{(m+1)(k-1)+1}^{n+1} - F_{(m+1)(k-1)+1}^{n+1} G_{(m+1)k+1}^n = -F_{(m+1)(k-1)+2}^n F_{(m+1)k}^{n+1},$$

and, for $i = 2, \dots, m + 1$,

$$(5.6) \quad F_{(m+1)k+i}^n G_{(m+1)k+i}^{n+1} - F_{(m+1)k+i}^{n+1} G_{(m+1)k+i}^n = F_{(m+1)k+i+1}^n F_{(m+1)k+i-1}^{n+1},$$

$$(5.7) \quad F_{(m+1)k+i}^n G_{(m+1)(k-1)+i}^{n+1} - F_{(m+1)(k-1)+i}^{n+1} G_{(m+1)k+i}^n = F_{(m+1)(k-1)+i+1}^n F_{(m+1)k+i-1}^{n+1},$$

then (5.2) follows.

Proof. Let us first notice that, taking $i = 1$ in (5.6) and (5.7) gives (5.4) and (5.5), respectively, after a change of signs of their right-hand sides.

Let us separate (5.2) into the product of two relations, and prove that each of the following formulae holds separately,

$$(5.8) \quad \prod_{i=1}^m (F_{k+i}^n G_{k+i}^{n+1} - F_{k+i}^{n+1} G_{k+i}^n) = \pm \prod_{i=1}^m F_{k+i+1}^n F_{k+i-1}^{n+1},$$

which are the products appearing, respectively, in the right and left sides of (5.2), and

$$(5.9) \quad F_{k+m+1}^n G_k^{n+1} - F_k^{n+1} G_{k+m+1}^n = \mp F_{k+1}^n F_{k+m}^{n+1},$$

which are its remaining parts. Then, multiplying together (5.8) and (5.9), we will obtain (5.2), but we must notice that the signs used in (5.8) and (5.9) have to be opposite.

Let us assume that (5.4)–(5.7) hold true. The proofs of the relations (5.8) and (5.9) have to be separated into three cases according to the value of k in (5.2).

- k replaced by $(m + 1)k$ in (5.8) and (5.9).
 - Multiplying together the relations (5.6) for $i = 2, \dots, m$, and then multiplying each of its sides by the corresponding side of (5.4) (which brings a change in the sign) proves (5.8), with the sign $-$, and k replaced by $(m + 1)k$.
 - Considering (5.7) for $i = m + 1$, we have

$$F_{(m+1)(k+1)}^n G_{(m+1)k}^{n+1} - F_{(m+1)k}^{n+1} G_{(m+1)(k+1)}^n = F_{(m+1)k+1}^n F_{(m+1)k+m}^{n+1},$$

which is (5.9) with the sign $+$, and k replaced by $(m + 1)k$.

- We get (5.2) by multiplying together the two relations.
- k replaced by $(m + 1)k + 1$ in (5.8) and (5.9).
 - We multiply together the relations (5.6) for $i = 2, \dots, m + 1$. By adding 1 to all the lower indexes, and making the product for $i = 1, \dots, m$, we obtain (5.8) with the sign $+$.
 - Considering (5.5) with k replaced by $k + 1$, we obtain (5.9), with the sign $-$, and k replaced by $(m + 1)k + 1$.
 - Multiplying together the two relations, we obtain (5.2).
- k replaced by $(m + 1)k + j$, for $j = 2, \dots, m$, in (5.8) and (5.9).

◦ Let $2 \leq j \leq m$ be fixed. Let us write that (5.6) holds with $2 \leq i + j \leq m + 1$ instead of i , that is, for $i = 1, \dots, m - j + 1$,

$$F_{(m+1)k+i+j}^n G_{(m+1)k+i+j}^{n+1} - F_{(m+1)k+i+j}^{n+1} G_{(m+1)k+i+j}^n = F_{(m+1)k+i+j+1}^n F_{(m+1)k+i+j-1}^{n+1}.$$

Multiply together these relations for $i = 1, \dots, m - j + 1$, and then, take their product for $i = m - j + 2, \dots, m$. It is mandatory to consider separately these two products because the relations satisfied by the quantities involved are different. When $i = m - j + 2$, the left-hand side of this expression becomes

$$F_{(m+1)k+m+2}^n G_{(m+1)k+m+2}^{n+1} - F_{(m+1)k+m+2}^{n+1} G_{(m+1)k+m+2}^n,$$

and its right-hand side is equal to $F_{(m+1)k+m+3}^n F_{(m+1)k+m+1}^{n+1}$, that is, respectively,

$$F_{(m+1)(k+1)+1}^n G_{(m+1)(k+1)+1}^{n+1} - F_{(m+1)(k+1)+1}^{n+1} G_{(m+1)(k+1)+1}^n,$$

and $F_{(m+1)(k+1)+2}^n F_{(m+1)(k+1)}^{n+1}$. Thus, by (5.4) with k replaced by $k + 1$, these two expressions are equal after changing the sign on one side.

For $i = m - j + 3$, we have

$$F_{(m+1)k+m+3}^n G_{(m+1)k+m+3}^{n+1} - F_{(m+1)k+m+3}^{n+1} G_{(m+1)k+m+3}^n = F_{(m+1)k+m+4}^n F_{(m+1)k+m+2}^{n+1},$$

that is,

$$F_{(m+1)(k+1)+2}^n G_{(m+1)(k+1)+2}^{n+1} - F_{(m+1)(k+1)+2}^{n+1} G_{(m+1)(k+1)+2}^n = F_{(m+1)(k+1)+3}^n F_{(m+1)(k+1)+1}^{n+1},$$

which is (5.6) with $k + 1$ instead of k . And so on until $i = m$.

Thus, in the products from $i = m - j + 2$ to m , the sign is changed in one, and only one, of the expressions due to (5.4), and we finally obtain (5.8) with the sign $-$.

◦ Considering (5.7) with k replaced by $k + 1$, we obtain

$$F_{(m+1)(k+1)+i}^n G_{(m+1)k+i}^{n+1} - F_{(m+1)k+i}^{n+1} G_{(m+1)(k+1)+i}^n = F_{(m+1)k+i+1}^n F_{(m+1)(k+1)+i-1}^{n+1},$$

which is (5.9), with the sign $+$, and k replaced by $(m + 1)k + j$.

◦ The product of the two relations gives (5.2).

Thus, (5.2) has now been proved for all possible values of k . □

Finally, we are able to prove the following.

Theorem 5.3. *The relation (5.2) holds with the F_k^n 's and the G_k^n 's given by the following relations, for $k = 0, 1, \dots$,*

$$(5.10) \quad F_{(m+1)(k-1)+i}^n = H_k(\Delta^i S_n), \quad i = 1, \dots, m + 1,$$

$$(5.11) \quad G_{(m+1)(k-1)+1}^n = H_{k-1}(\Delta^{m+2} S_n), \quad G_{(m+1)k}^n = H_{k+1}(S_n),$$

$$(5.12) \quad G_{(m+1)(k-1)+i}^n = \Phi_{k+1}(\Delta^{i-1} S_n), \quad i = 2, \dots, m.$$

Proof. We will first prove (5.4)–(5.7), with the F_k^n 's and G_k^n 's given by (5.10)–(5.12).

Replacing the determinants in (4.1) by their expressions, we obtain

$$F_{(m+1)(k+1)}^n G_{(m+1)(k+1)}^{n+1} - F_{(m+1)(k+1)}^{n+1} G_{(m+1)(k+1)}^n = F_{(m+1)(k+1)+1}^n F_{(m+1)k+m}^{n+1}$$

which corresponds to (5.6) for the case $i = m + 1$. Replacing the determinants in (4.8) by their expressions, we obtain the bilinear equation (5.6) for the cases $i = 2, 3, \dots, m$, which completes the proof of the equation (5.6).

Replacing the determinants in (4.1) and (4.4), both with ΔS_n instead of S_n , by their expressions, we see that the equations (5.4) and (5.5) are satisfied.

Then, replacing the determinants in (4.4) by their expressions, we obtain

$$F_{(m+1)k}^n G_{(m+1)(k-1)}^{n+1} - F_{(m+1)(k-1)}^{n+1} G_{(m+1)k}^n = F_{(m+1)(k-1)+1}^n F_{(m+1)(k-1)+m}^{n+1}$$

which corresponds to (5.7) for the case $i = m + 1$, while replacing the determinants in (4.7) by their expressions, leads to the bilinear equation (5.7) for $i = 2, 3, \dots, m$, which completes the proof for the equation (5.7).

Since the identities (5.4)–(5.7) hold, then, by Lemma 5.2, (5.2) follows with the F_k^n 's and the G_k^n 's given by (5.10)–(5.12). \square

We also have the following result which completes the bunch of equalities.

Corollary 5.4.

$$F_{k+m+1}^n F_{k-1}^{n+1} = F_k^n F_{k+m}^{n+1} - F_{k+m}^n F_k^{n+1}.$$

Proof. Replacing the determinants in (4.3) by their expressions given by (5.10), and k by $k + 1$, we obtain, for $i = 1, \dots, m$, the following relation without any G_k^n :

$$F_{(m+1)k+i+1}^n F_{(m+1)(k-1)+i}^{n+1} = F_{(m+1)(k-1)+i+1}^n F_{(m+1)k+i}^{n+1} - F_{(m+1)k+i}^n F_{(m+1)(k-1)+i+1}^{n+1}.$$

Similarly, the determinantal identity (4.9) leads, after replacing k by $k + 2$, to

$$F_{(m+1)(k+1)+1}^n F_{(m+1)k}^{n+1} = F_{(m+1)k+1}^n F_{(m+1)(k+1)}^{n+1} - F_{(m+1)(k+1)}^n F_{(m+1)k+1}^{n+1},$$

which is the preceding relation for $i = m + 1$. Thus, changing mk into m , these two identities can be gathered into the single formula of the corollary. \square

6. THE LINK BETWEEN THE MULTISTEP SHANKS TRANSFORMATION AND THE MULTISTEP ε -ALGORITHM

Thanks to the results obtained in Section 5, by using Hirota's bilinear method, we proved (Theorem 5.3) that, if G_k^n and F_k^n are given by (5.10)–(5.12), then (5.2) follows. Thus, we are now able to give the determinantal formulae for the quantities computed by the multistep ε -algorithm (5.1) and we have the following.

Theorem 6.1. *The quantities $\varepsilon_{k,m}^{(n)}$ computed by the multistep ε -algorithm (3.2), with the initializations (3.3), are expressed by the ratios of determinants (3.4), (3.6), and (3.7).*

Consequently,

Corollary 6.2. *The quantities $\varepsilon_{(m+1)k,m}^{(n)}$ computed by (3.2) are expressed by the ratio of determinants (3.4).*

Thus, by following the approach named *Proof*, as explained in Section 2, starting from the knowledge of the algorithm (3.2), we were able to find a ratio of determinants for the transformation it implements (the multistep Shanks transformation (3.1)).

When $m = 1$, the relations (3.7) disappear, and Hirota's bilinear method leads to a new proof that the ε -algorithm of Wynn implements the Shanks sequence transformation and, that the quantities computed by this algorithm are expressed by the ratios of Hankel determinants defining the Shanks transformation.

We will now verify the identity between the recursive rule (3.2) of the multistep ε -algorithm, the ratio of determinants (3.4) (the multistep Shanks transformation), and the additional ratios of determinants appearing in (3.6) and (3.7), by using only the relations between determinants given in Section 4.

We start by considering the quantities computed by the main relation (3.4). From the determinantal identity (4.4), we get

$$\begin{aligned}
 \varepsilon_{(m+1)k,m}^{(n)} - \varepsilon_{(m+1)(k-1),m}^{(n+1)} &= \frac{H_{k+1}(S_n)}{H_k(\Delta^{m+1}S_n)} - \frac{H_k(S_{n+1})}{H_{k-1}(\Delta^{m+1}S_{n+1})} \\
 &= \frac{H_{k+1}(S_n)H_{k-1}(\Delta^{m+1}S_{n+1}) - H_k(S_{n+1})H_k(\Delta^{m+1}S_n)}{H_k(\Delta^{m+1}S_n)H_{k-1}(\Delta^{m+1}S_{n+1})} \\
 (6.1) \quad &= -\frac{H_k(\Delta S_n)H_k(\Delta^m S_{n+1})}{H_k(\Delta^{m+1}S_n)H_{k-1}(\Delta^{m+1}S_{n+1})}.
 \end{aligned}$$

We have now to show that

$$(6.2) \quad \varepsilon_{(m+1)k,m}^{(n)} - \varepsilon_{(m+1)(k-1),m}^{(n+1)} = 1 / \prod_{i=1}^m (\varepsilon_{(m+1)(k-1)+i,m}^{(n+1)} - \varepsilon_{(m+1)(k-1)+i,m}^{(n)}).$$

In fact, from the identity (4.1), with ΔS_n instead of S_n , and by using the relation (3.6), we get

$$\begin{aligned}
 \varepsilon_{(m+1)(k-1)+1,m}^{(n+1)} - \varepsilon_{(m+1)(k-1)+1,m}^{(n)} &= \frac{H_{k-1}(\Delta^{m+2}S_{n+1})}{H_k(\Delta S_{n+1})} - \frac{H_{k-1}(\Delta^{m+2}S_n)}{H_k(\Delta S_n)} \\
 &= \frac{H_{k-1}(\Delta^{m+2}S_{n+1})H_k(\Delta S_n) - H_{k-1}(\Delta^{m+2}S_n)H_k(\Delta S_{n+1})}{H_k(\Delta S_{n+1})H_k(\Delta S_n)} \\
 (6.3) \quad &= -\frac{H_k(\Delta^2 S_n)H_{k-1}(\Delta^{m+1}S_{n+1})}{H_k(\Delta S_{n+1})H_k(\Delta S_n)},
 \end{aligned}$$

and, from the identity (4.8) and the relation (3.7), we have, for $i = 2, \dots, m$,

$$\begin{aligned}
 \varepsilon_{(m+1)(k-1)+i,m}^{(n+1)} - \varepsilon_{(m+1)(k-1)+i,m}^{(n)} &= \frac{\Phi_{k+1}(\Delta^{i-1}S_{n+1})}{H_k(\Delta^i S_{n+1})} - \frac{\Phi_{k+1}(\Delta^{i-1}S_n)}{H_k(\Delta^i S_n)} \\
 &= \frac{\Phi_{k+1}(\Delta^{i-1}S_{n+1})H_k(\Delta^i S_n) - \Phi_{k+1}(\Delta^{i-1}S_n)H_k(\Delta^i S_{n+1})}{H_k(\Delta^i S_{n+1})H_k(\Delta^i S_n)} \\
 (6.4) \quad &= \frac{H_k(\Delta^{i+1}S_n)H_k(\Delta^{i-1}S_{n+1})}{H_k(\Delta^i S_{n+1})H_k(\Delta^i S_n)}.
 \end{aligned}$$

Thus, using (6.3) and (6.4), and simplifying the terms in the numerators and denominators we obtain

$$\begin{aligned}
 & \prod_{i=1}^m (\varepsilon_{(m+1)(k-1)+i,m}^{(n+1)} - \varepsilon_{(m+1)(k-1)+i,m}^{(n)}) \\
 &= - \frac{H_k(\Delta^2 S_n) H_{k-1}(\Delta^{m+1} S_{n+1})}{H_k(\Delta S_{n+1}) H_k(\Delta S_n)} \prod_{i=2}^m \frac{H_k(\Delta^{i+1} S_n) H_k(\Delta^{i-1} S_{n+1})}{H_k(\Delta^i S_n) H_k(\Delta^i S_{n+1})} \\
 (6.5) \quad &= - \frac{H_{k-1}(\Delta^{m+1} S_{n+1}) H_k(\Delta^{m+1} S_n)}{H_k(\Delta S_n) H_k(\Delta^m S_{n+1})}.
 \end{aligned}$$

Comparing (6.1) and (6.5), we obtain (6.2), that is the rule (3.2) of the multistep ε -algorithm with the lower index k replaced by $(m + 1)k - 1$.

We consider now the quantities computed by the intermediate relation (3.6). Again from the determinantal identity (4.4), with S_n replaced by ΔS_n , we get

$$\begin{aligned}
 \varepsilon_{(m+1)(k-1)+1,m}^{(n)} - \varepsilon_{(m+1)(k-2)+1,m}^{(n+1)} &= \frac{H_{k-1}(\Delta^{m+2} S_n)}{H_k(\Delta S_n)} - \frac{H_{k-2}(\Delta^{m+2} S_{n+1})}{H_{k-1}(\Delta S_{n+1})} \\
 &= \frac{H_{k-1}(\Delta^{m+2} S_n) H_{k-1}(\Delta S_{n+1}) - H_{k-2}(\Delta^{m+2} S_{n+1}) H_k(\Delta S_n)}{H_k(\Delta S_n) H_{k-1}(\Delta S_{n+1})} \\
 (6.6) \quad &= \frac{H_{k-1}(\Delta^2 S_n) H_{k-1}(\Delta^{m+1} S_{n+1})}{H_k(\Delta S_n) H_{k-1}(\Delta S_{n+1})}.
 \end{aligned}$$

We now have to show that

$$\begin{aligned}
 & \varepsilon_{(m+1)(k-1)+1,m}^{(n)} - \varepsilon_{(m+1)(k-2)+1,m}^{(n+1)} = \\
 (6.7) \quad & 1 / \prod_{i=1}^m (\varepsilon_{(m+1)(k-2)+1+i,m}^{(n+1)} - \varepsilon_{(m+1)(k-2)+1+i,m}^{(n)}).
 \end{aligned}$$

From (4.1), and by using (3.4), we get

$$\begin{aligned}
 \varepsilon_{(m+1)(k-1),m}^{(n+1)} - \varepsilon_{(m+1)(k-1),m}^{(n)} &= \frac{H_k(S_{n+1})}{H_{k-1}(\Delta^{m+1} S_{n+1})} - \frac{H_k(S_n)}{H_{k-1}(\Delta^{m+1} S_n)} \\
 &= \frac{H_k(S_{n+1}) H_{k-1}(\Delta^{m+1} S_n) - H_k(S_n) H_{k-1}(\Delta^{m+1} S_{n+1})}{H_{k-1}(\Delta^{m+1} S_{n+1}) H_{k-1}(\Delta^{m+1} S_n)} \\
 (6.8) \quad &= \frac{H_k(\Delta S_n) H_{k-1}(\Delta^m S_{n+1})}{H_{k-1}(\Delta^{m+1} S_{n+1}) H_{k-1}(\Delta^{m+1} S_n)}.
 \end{aligned}$$

We can also derive the following formula from (6.8) and (6.4) after simplifying the terms in the numerators and denominators:

$$\begin{aligned}
& \prod_{i=2}^{m+1} (\varepsilon_{(m+1)(k-2)+i,m}^{(n+1)} - \varepsilon_{(m+1)(k-2)+i,m}^{(n)}) \\
&= \frac{H_k(\Delta S_n)H_{k-1}(\Delta^m S_{n+1})}{H_{k-1}(\Delta^{m+1} S_{n+1})H_{k-1}(\Delta^{m+1} S_n)} \prod_{i=2}^m \frac{H_{k-1}(\Delta^{i+1} S_n)H_{k-1}(\Delta^{i-1} S_{n+1})}{H_{k-1}(\Delta^i S_n)H_{k-1}(\Delta^i S_{n+1})} \\
(6.9) \quad &= \frac{H_k(\Delta S_n)H_{k-1}(\Delta S_{n+1})}{H_{k-1}(\Delta^2 S_n)H_{k-1}(\Delta^{m+1} S_{n+1})} \\
&= \prod_{i=1}^m (\varepsilon_{(m+1)(k-2)+1+i,m}^{(n+1)} - \varepsilon_{(m+1)(k-2)+1+i,m}^{(n)}).
\end{aligned}$$

Comparing (6.6) and (6.9), we obtain (6.7), that is the rule (3.2) of the multistep ε -algorithm where the lower index k is replaced by $(m+1)(k-1)$.

Finally, we consider the relation (3.7). From the identity (4.7), we get the following relation:

$$\begin{aligned}
\varepsilon_{(m+1)(k-1)+i,m}^{(n)} - \varepsilon_{(m+1)(k-2)+i,m}^{(n+1)} &= \frac{\Phi_{k+1}(\Delta^{i-1} S_n)}{H_k(\Delta^i S_n)} - \frac{\Phi_k(\Delta^{i-1} S_{n+1})}{H_{k-1}(\Delta^i S_{n+1})} \\
&= \frac{\Phi_{k+1}(\Delta^{i-1} S_n)H_{k-1}(\Delta^i S_{n+1}) - \Phi_k(\Delta^{i-1} S_{n+1})H_k(\Delta^i S_n)}{H_k(\Delta^i S_n)H_{k-1}(\Delta^i S_{n+1})} \\
(6.10) \quad &= -\frac{H_k(\Delta^{i-1} S_{n+1})H_{k-1}(\Delta^{i+1} S_n)}{H_k(\Delta^i S_n)H_{k-1}(\Delta^i S_{n+1})}, \quad i = 2, \dots, m.
\end{aligned}$$

We have to show that, for $i = 2, \dots, m$,

$$\begin{aligned}
& \varepsilon_{(m+1)(k-1)+i,m}^{(n)} - \varepsilon_{(m+1)(k-2)+i,m}^{(n+1)} = \\
(6.11) \quad & 1 / \prod_{j=1}^m (\varepsilon_{(m+1)(k-2)+i+j,m}^{(n+1)} - \varepsilon_{(m+1)(k-2)+i+j,m}^{(n)}).
\end{aligned}$$

By using (6.8), (6.4) and (6.8), we have, for $i = 2, \dots, m$,

$$\begin{aligned}
& \prod_{j=i+1}^{m+i} (\varepsilon_{(m+1)(k-2)+j,m}^{(n+1)} - \varepsilon_{(m+1)(k-2)+j,m}^{(n)}) \\
&= (\varepsilon_{(m+1)(k-1),m}^{(n+1)} - \varepsilon_{(m+1)(k-1),m}^{(n)}) \prod_{j=i+1}^m (\varepsilon_{(m+1)(k-2)+j,m}^{(n+1)} - \varepsilon_{(m+1)(k-2)+j,m}^{(n)}) \\
&\quad \times (\varepsilon_{(m+1)(k-1)+1,m}^{(n+1)} - \varepsilon_{(m+1)(k-1)+1,m}^{(n)}) \prod_{j=2}^{i-1} (\varepsilon_{(m+1)(k-1)+j,m}^{(n+1)} - \varepsilon_{(m+1)(k-1)+j,m}^{(n)}) \\
&= - \prod_{j=i+1}^m \frac{H_{k-1}(\Delta^{j+1} S_n)H_{k-1}(\Delta^{j-1} S_{n+1})}{H_{k-1}(\Delta^j S_{n+1})H_{k-1}(\Delta^j S_n)} \prod_{j=2}^{i-1} \frac{H_k(\Delta^{j+1} S_n)H_k(\Delta^{j-1} S_{n+1})}{H_k(\Delta^j S_{n+1})H_k(\Delta^j S_n)} \\
&\quad \times \frac{H_k(\Delta S_n)H_{k-1}(\Delta^m S_{n+1})}{H_{k-1}(\Delta^{m+1} S_{n+1})H_{k-1}(\Delta^{m+1} S_n)} \frac{H_k(\Delta^2 S_n)H_{k-1}(\Delta^{m+1} S_{n+1})}{H_k(\Delta S_{n+1})H_k(\Delta S_n)}
\end{aligned}$$

$$\begin{aligned}
 (6.12) \quad &= -\frac{H_{k-1}(\Delta^i S_{n+1})H_k(\Delta^i S_n)}{H_{k-1}(\Delta^{i+1} S_n)H_k(\Delta^{i-1} S_{n+1})} \\
 &= \prod_{j=1}^m (\varepsilon_{(m+1)(k-2)+i+j,m}^{(n+1)} - \varepsilon_{(m+1)(k-2)+i+j,m}^{(n)}).
 \end{aligned}$$

Comparing (6.10) and (6.12), we obtain (6.11), that is the rule (3.2) of the multistep ε -algorithm where the lower index k is replaced by $(m + 1)(k - 1) + i - 1$, for $i = 2, \dots, m$.

Thus, following the approach named *Verification*, as explained in Section 2, we showed the equivalence between the ratio of determinants defining the multistep Shanks transformation (3.1) and the additional ratios of determinants appearing in the right sides of (3.6) and (3.7), and the multistep ε -algorithm (3.2).

7. PROPERTIES OF THE MULTISTEP SHANKS TRANSFORMATION

As already stated in Section 2, the E -transformation is a very general sequence transformation covering many known transformations. It is given under the form of the following ratio of determinants

$$E_k(S_n) = \frac{\begin{vmatrix} S_n & S_{n+1} & \cdots & S_{n+k} \\ g_1(n) & g_1(n+1) & \cdots & g_1(n+k) \\ \vdots & \vdots & & \vdots \\ g_k(n) & g_k(n+1) & \cdots & g_k(n+k) \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ g_1(n) & g_1(n+1) & \cdots & g_1(n+k) \\ \vdots & \vdots & & \vdots \\ g_k(n) & g_k(n+1) & \cdots & g_k(n+k) \end{vmatrix}}$$

and it can be implemented by a recursive algorithm called the E -algorithm whose rules are

$$\begin{aligned}
 E_k^{(n)} &= \frac{g_{k-1,k}^{(n+1)} E_{k-1}^{(n)} - g_{k-1,k}^{(n)} E_{k-1}^{(n+1)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}}, \\
 g_{k,i}^{(n)} &= \frac{g_{k-1,k}^{(n+1)} g_{k-1,i}^{(n)} - g_{k-1,k}^{(n)} g_{k-1,i}^{(n+1)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}},
 \end{aligned}$$

with $E_0^{(n)} = S_n$ and $g_{0,i}^{(n)} = g_i(n)$ for $n = 0, 1, \dots$

Comparing this ratio with (3.1), we see that the multistep Shanks transformation corresponds to the choice $g_i(n) = \Delta^{im} S_n$ for $i = 1, 2, \dots$, and for all n , for the auxiliary quantities involved in the E -transformation and the E -algorithm.

Thus, the multistep ε -algorithm is not the only algorithm for implementing the multistep Shanks transformation (3.1). It can likewise be implemented by the E -algorithm, and we get, for all k and n ,

$$E_k^{(n)} = e_{k,m}(S_n).$$

Since the numerator and the denominator in (5.1) are not defined uniquely but up to a common multiplying factor, it should also be possible to implement the multistep Shanks transformation by an algorithm generalizing the Ψ -algorithm of Ford and Sidi [13].

Thanks to the connection with the E -transformation, all the algebraic properties of this transformation, and all the general convergence and acceleration results proved for it, for example in [6, 22], also hold for the multistep Shanks transformation. In particular, from the fundamental property of the E -transformation, or directly from (3.1), we see that the *kernel* of the multistep Shanks transformation (that is the set of sequences which are transformed into a constant sequence) is given by the following.

Theorem 7.1. *A necessary and sufficient condition that, for all n , $e_{k,m}(S_n) = S$ is that there exist constants a_1, \dots, a_k , $a_k \neq 0$, such that, for all n ,*

$$S_n = S + a_1 \Delta^m S_n + a_2 \Delta^{2m} S_n + \dots + a_k \Delta^{km} S_n.$$

Let us recall that the kernel of the original Shanks transformation $e_{km} : (S_n) \mapsto (e_{km}(S_n) = \varepsilon_{2km}^{(n)})$ is the set of sequences such that, for all n , $S_n = S + b_1 \Delta S_n + \dots + b_{km} \Delta^{km} S_n$, where b_1, \dots, b_{km} , $b_{km} \neq 0$, are constants. Thus, we have the following.

Corollary 7.2. *The kernel of the multistep Shanks transformation $e_{k,m}$ is contained into the kernel of the Shanks transformation e_{km} .*

From the preceding results, another interesting property is that, as noticed in [2] and fully explained in [7], we have

$$\begin{aligned} \varepsilon_{(m+1)k,m}^{(n)} &= \frac{f_{k,m}(S_n, \dots, S_{n+(m+1)k})}{Df_{k,m}(S_n, \dots, S_{n+(m+1)k})}, \\ \varepsilon_{(m+1)k+1,m}^{(n)} &= \frac{Df_{k,m}(\Delta S_n, \dots, \Delta S_{n+(m+1)k})}{f_{k,m}(\Delta S_n, \dots, \Delta S_{n+(m+1)k})}, \end{aligned}$$

where $f_{k,m}$ is a function depending on $(m+1)k+1$ variables and such that $D^2 f_{k,m} \equiv 0$, where $Df_{k,m}$ denotes the sum of the partial derivatives of $f_{k,m}$. Thus, we obtain the following connection with Hirota's bilinear method,

$$\begin{aligned} G_{(m+1)k}^n &= f_{k,m}(S_n, \dots, S_{n+(m+1)k}), \\ F_{(m+1)k}^n &= Df_{k,m}(S_n, \dots, S_{n+(m+1)k}), \\ G_{(m+1)k+1}^n &= Df_{k,m}(\Delta S_n, \dots, \Delta S_{n+(m+1)k}), \\ F_{(m+1)k+1}^n &= f_{k,m}(\Delta S_n, \dots, \Delta S_{n+(m+1)k}), \end{aligned}$$

and, according to this theory, the multistep Shanks transformation is quasilinear, that is, $e_{k,m}(aS_n + b) = ae_{k,m}(S_n) + b$, a result which can be seen directly from (3.1).

8. AN EXTENDED DISCRETE LOTKA-VOLTERRA SYSTEM

Recently, as explained in Section 1, it has been shown that integrable systems are closely related to numerical algorithms. On one hand, some numerical algorithms are found to be soliton equations. For example, one step of the QR -algorithm is equivalent to the time evolution of the finite nonperiodic Toda lattice [40]. The ε -algorithm is nothing but the fully-discrete potential KdV equation, and the ρ -algorithm is considered to be the fully-discrete cylindrical KdV equations or the Milne-Thomson equation; see [46, 32, 25, 27, 42]. On the other hand, integrable systems can be used for designing new numerical algorithms. For example, the discrete Lotka-Volterra system has applications in numerical algorithms for computing

singular values [41, 20, 21], the continuous-time Toda equation leads to a new algorithm for computing the Laplace transform of a given analytic function [28], and the discrete relativistic Toda molecule equation leads to a new Padé approximation algorithm for formal power series [24].

In this section, we will show that there exist a *Miura transformation* between the multistep ε -algorithm (3.2) and a discrete integrable system. In fact, if we set $\left(a_{k-\frac{m-1}{2}}^{(n)}\right)^{-1} = \varepsilon_{k,m}^{(n+1)} - \varepsilon_{k,m}^{(n)}$, then equation (3.2) is transformed into the *extended discrete Lotka-Volterra equation*:

$$(8.1) \quad \prod_{i=0}^{m-1} a_{k-\frac{m-1}{2}+i}^{(n+1)} - \prod_{i=0}^{m-1} a_{k-\frac{m-1}{2}+i}^{(n)} = \frac{1}{a_{k+\frac{m+1}{2}}^{(n)}} - \frac{1}{a_{k-\frac{m+1}{2}}^{(n+1)}}.$$

This equation can be considered as the time discretization, for $N = -1$, of

$$(8.2) \quad \frac{d}{dt} \left(\prod_{i=0}^{m-1} a_{k-\frac{m-1}{2}+i} \right) = \prod_{i=0}^{-N-1} a_{k+\frac{m+1}{2}+i}^{-1} - \prod_{i=0}^{-N-1} a_{k-\frac{m+1}{2}-i}^{-1},$$

$$m = 1, 2, \dots, \quad N = -1, -2, \dots,$$

which is called the *extended Lotka-Volterra equation*. This equation was first proposed in [29], and it was developed in [19]. Indeed, with $N = -1$, (8.2) becomes

$$(8.3) \quad \frac{d}{dt} \left(\prod_{i=0}^{m-1} a_{k-\frac{m-1}{2}+i} \right) = \frac{1}{a_{k+\frac{m+1}{2}}} - \frac{1}{a_{k-\frac{m+1}{2}}}.$$

Now, consider n as the discretization of t , and replace the derivative in the left-hand side of (8.3) by the forward difference Δ acting on n . The left-hand side becomes

$$\prod_{i=0}^{m-1} a_{k-\frac{m-1}{2}+i}^{(n+1)} - \prod_{i=0}^{m-1} a_{k-\frac{m-1}{2}+i}^{(n)}.$$

Then, replace a_k in the first term of the right-hand side of (8.3) by $a_k^{(n)}$, and, in its second term, by $a_k^{(n+1)}$. We get (8.1).

Using the relations (5.4) and (5.6), we obtain the solution of (8.1)

$$a_{(m+1)k+1-\frac{m-1}{2}}^{(n)} = -\frac{1}{\varepsilon_{(m+1)k+1,m}^{(n+1)} - \varepsilon_{(m+1)k+1,m}^{(n)}} = -\frac{F_{(m+1)k+1}^n F_{(m+1)k+1}^{n+1}}{F_{(m+1)k+2}^n F_{(m+1)k}^{n+1}},$$

$$a_{(m+1)k+i-\frac{m-1}{2}}^{(n)} = \frac{1}{\varepsilon_{(m+1)k+i,m}^{(n+1)} - \varepsilon_{(m+1)k+i,m}^{(n)}} = \frac{F_{(m+1)k+i}^n F_{(m+1)k+i}^{n+1}}{F_{(m+1)k+i+1}^n F_{(m+1)k+i-1}^{n+1}},$$

$$i = 2, \dots, m + 1,$$

that is,

$$(8.4) \quad a_{(m+1)k-\frac{m-1}{2}}^{(n)} = \frac{H_k(\Delta^{m+1} S_n) H_k(\Delta^{m+1} S_{n+1})}{H_{k+1}(\Delta S_n) H_k(\Delta^m S_{n+1})},$$

$$(8.5) \quad a_{(m+1)k-\frac{m-1}{2}+1}^{(n)} = -\frac{H_{k+1}(\Delta S_n) H_{k+1}(\Delta S_{n+1})}{H_{k+1}(\Delta^2 S_n) H_k(\Delta^{m+1} S_{n+1})},$$

$$(8.6) \quad a_{(m+1)k-\frac{m-1}{2}+j}^{(n)} = \frac{H_{k+1}(\Delta^j S_n) H_{k+1}(\Delta^j S_{n+1})}{H_{k+1}(\Delta^{j+1} S_n) H_{k+1}(\Delta^{j-1} S_{n+1})},$$

for $j = 2, \dots, m$, and $k = -m + 1, -m + 2, \dots$, with the initial values

$$(8.7) \quad a_{-m-\frac{m-1}{2}}^{(n)} = \infty, a_{-m+1-\frac{m-1}{2}}^{(n)} = \dots = a_{-1-\frac{m-1}{2}}^{(n)} = n, a_{-\frac{m-1}{2}}^{(n)} = 1/\Delta S_n.$$

The difference equation (8.1), with the initial values (8.7), is said to be the integrable time discretization of the extended Lotka–Volterra equation (8.2) in the sense that its solution is given by (8.4)–(8.6). Conversely, the extended discrete Lotka–Volterra equation (8.2) can be seen as the time continuation of (8.1) with the initializations (8.7).

Consider the particular case $m = 1$. Then, (3.2) reduces to the ε -algorithm and equation (8.1) becomes

$$a_k^{(n+1)} - a_k^{(n)} = \frac{1}{a_{k+1}^{(n)}} - \frac{1}{a_{k-1}^{(n+1)}}.$$

By the dependent variable transformation

$$\frac{u_k^{(n)}}{u_{k-1}^{(n+1)}} = \frac{a_{k-1}^{(n+1)}}{a_{k+1}^{(n)}},$$

we obtain the discrete Lotka–Volterra equation

$$(8.8) \quad u_k^{(n+1)} \left(1 + u_{k-1}^{(n+1)}\right) = u_k^{(n)} \left(1 + u_{k+1}^{(n)}\right).$$

Then, the ε -algorithm can be transformed into the discrete Lotka–Volterra equation (8.8) through the following Miura transformation:

$$\frac{u_k^{(n)}}{u_{k-1}^{(n+1)}} = \frac{\varepsilon_{k+1,1}^{(n+1)} - \varepsilon_{k+1,1}^{(n)}}{\varepsilon_{k-1,1}^{(n+2)} - \varepsilon_{k-1,1}^{(n+1)}}.$$

Thus, the ε -algorithm can be considered as the discrete Lotka–Volterra equation (8.8), and more generally, the multistep ε -algorithm (3.2) is equivalent to the extended discrete Lotka–Volterra equation (8.1).

9. CONCLUSION AND FUTURE RESEARCH

Starting from the recursive rule (3.2) of the multistep ε -algorithm, by using Hirota's bilinear method, we were able to *prove* that the $\varepsilon_{k,m}^{(n)}$'s can be defined as ratios of determinants, and, as a consequence, that this algorithm implements the multistep Shanks transformation. The difficult point was to find to which determinants this identity had to be applied. We were also able to *verify* the equivalence between the recursive rule (3.2) of the multistep ε -algorithm and the determinantal formulae (3.4), (3.6), and (3.7), that define the quantities $\varepsilon_{k,m}^{(n)}$. It must be noticed that, contrary to the approaches of [46] and [15], we did not make use of Schweins' determinantal identity, but only of Sylvester's. Then, we showed that the multistep ε -algorithm was related to an extended discrete Lotka–Volterra system.

The approach developed above could possibly be extended to other nonlinear convergence acceleration algorithms such as, for example, the q -difference version of the ε -algorithm proposed in [16], or its two generalizations given in [5], or the other one presented in [17], or the general ε -algorithm of [12], or the ρ -algorithm [47], and the γ -algorithm which generalizes it [3]. Other algorithms related to them, such as the qd , the η , the ω , and the rs -algorithms, and the g -decomposition,

could also possibly be treated in a similar way (see [11] for their definitions). The quantities computed by these algorithms are all defined as ratios of determinants. These extensions, as well as extensions to other acceleration algorithms, will be the subject of future works. Let us mention that the confluent form of the multistep ε -algorithm is studied in [10]. It leads to a multistep Lotka–Volterra equation.

ACKNOWLEDGEMENTS

This work was partially supported by the National Natural Science Foundation of China (Grant no. 11071241), and the knowledge innovation program of LSEC and the Institute of Computational Math., AMSS, CAS. C. Brezinski would like to thank X.B. Hu, the State Key Laboratory of Scientific and Engineering Computing (LSEC), and the Institute of Computational Mathematics, AMSS, CAS, for inviting him for a stay during which part of this work was done. The work of Michela Redivo–Zaglia was partially supported by MIUR, PRIN grant no. 20083KLJEZ-003, and by the University of Padova, Project 2008 no. CPDA089040. We thank the reviewer for pertinent comments which helped us to improve the overall presentation of our results.

REFERENCES

- [1] A.C. Aitken, *Determinants and Matrices*, Oliver and Boyd, Edinburgh and London, 1939. MR00000217 (1:35a)
- [2] M.D. Benchiboun, *Étude de Certaines Généralisations du Δ^2 d'Aitken et Comparaison de Procédés d'Accélération de la Convergence*, Thèse de 3ème Cycle, Université des Sciences et Techniques de Lille, 1987.
- [3] C. Brezinski, *Méthodes d'Accélération de la Convergence en Analyse Numérique*, Thèse d'État, Université Scientifique et Médicale de Grenoble, 1971.
- [4] C. Brezinski, Accélération de suites à convergence logarithmique, C. R. Acad. Sci. Paris, Sér. A, 273 (1971) 727–730. MR0305544 (46:4674)
- [5] C. Brezinski, Conditions d'application et de convergence de procédés d'extrapolation, Numer. Math., 20 (1972) 64–79. MR0378341 (51:14509)
- [6] C. Brezinski, A general extrapolation algorithm, Numer. Math., 35 (1980) 175–187. MR585245 (81j:65015)
- [7] C. Brezinski, Quasi-linear extrapolation processes, in *Numerical Mathematics. Singapore 1988*, R.P. Agarwal et al. eds., ISNM vol. 86, Birkhäuser, Basel, 1988, pp. 61–78. MR1022946 (90k:65012)
- [8] C. Brezinski, *Biorthogonality and its Applications of Numerical Analysis*, Marcel Dekker, New York, 1992. MR1151616 (92m:41001)
- [9] C. Brezinski, Cross rules and non-Abelian lattice equations for the discrete and confluent non-scalar epsilon-algorithms, J. Phys. A: Math. Theor., 43 (2010) 205201. MR2639912 (2011d:65438)
- [10] C. Brezinski, Y. He, X.–B. Hu, J.–Q. Sun, H.–W. Tam, Confluent form of the multistep ε -algorithm, and the relevant integrable system, Stud. Appl. Math., 127 (2011), 191–209. DOI:10.1111/j.1467-9590.2011.00518.x
- [11] C. Brezinski, M. Redivo–Zaglia, *Extrapolation Methods. Theory and Practice*, North–Holland, Amsterdam, 1991. MR1140920 (93d:65001)
- [12] C. Carstensen, On a general epsilon algorithm, in *Numerical and Applied Mathematics*, C. Brezinski ed., Baltzer, Basel, 1989, pp. 437–441. MR1066038 (91g:65010)
- [13] W. F. Ford, A. Sidi, An algorithm for a generalization of the Richardson extrapolation process, SIAM J. Numer. Anal., 24 (1987) 1212–1232. MR909075 (89a:65006)
- [14] T. Hävie, Generalized Neville type extrapolation schemes, BIT, 19 (1979) 204–213. MR537780 (80f:65005)
- [15] Y. He, X.–B. Hu, J.–Q. Sun, E.J. Weniger, Convergence acceleration algorithm via an equation related to Boussinesq equation, SIAM J. Sci. Comput., 33 (2011), 1234–1245.

- [16] Y. He, X.-B. Hu, H.-W. Tam, A q -difference version of the ε -algorithm, *J. Phys. A: Math. Theor.*, 42 (2009) 095202. MR2525530 (2010j:37111)
- [17] Y. He, X.-B. Hu, H.-W. Tam, S. Tsujimoto, Convergence acceleration algorithms related to a general E -transformation and its particular cases, submitted to *Japan J. Indust. Appl. Math.*
- [18] R. Hirota, *The Direct Method in Soliton Theory*, Cambridge University Press, Cambridge, 1992. MR2085332 (2005f:37137)
- [19] X.-B. Hu, R.K. Bullough, Bäcklund transformation and nonlinear superposition formula of an extended Lotka–Volterra equation, *J. Phys. A: Math. Gen.*, 30 (1997) 3635–3641. MR1457967 (98i:35159)
- [20] M. Iwasaki, Y. Nakamura, On the convergence of a solution of the discrete Lotka–Volterra system, *Inverse Problem*, 18 (2002) 1569–1578. MR1955905 (2004a:39031)
- [21] M. Iwasaki, Y. Nakamura, An application of the discrete Lotka–Volterra system with variable step-size to singular value computation, *Inverse Problems* 20 (2004) 553–563. MR2065439 (2005e:39046)
- [22] A. Matos, M. Prévost, Acceleration property for the columns of the E -algorithm, *Numer. Algorithms*, 2 (1992) 393–408. MR1184823 (94b:65008)
- [23] G. Meinardus, G.D. Taylor, Lower estimates for the error of the best uniform approximation, *J. Approx. Theory*, 16 (1976) 150–161. MR0399720 (53:3562)
- [24] Y. Minesaki, Y. Nakamura, The discrete relativistic Toda molecule equation and a Padé approximation algorithm, *Numer. Algorithms*, 27 (2001) 219–235. MR1858990 (2002h:42054)
- [25] A. Nagai, J. Satsuma, Discrete soliton equations and convergence acceleration algorithms, *Phys. Letters A*, 209 (1995) 305–312. MR1368590 (96h:65006)
- [26] A. Nagai, T. Tokihiro, J. Satsuma, The Toda molecule equation and the ε -algorithm, *Math. Comput.*, 67 (1998) 1565–1575. MR1484902 (99d:65010)
- [27] Y. Nakamura, ed., *Applied Integrable Systems* (in Japanese), Syokabo, Tokyo, 2000.
- [28] Y. Nakamura, Calculating Laplace transforms in terms of the Toda molecule, *SIAM J. Sci. Comput.* 20 (1999) 306–317. MR1639142 (2000i:65215)
- [29] K. Narita, Soliton solution to extended Volterra equation, *J. Phys. Soc. Japan*, 51 (1982) 1682–1685. MR668131 (84c:34116)
- [30] F.W. Nijhoff, Discrete Painlevé equations and symmetry reductions on the lattice, in *Discrete Integrable Geometry and Physics*, A. Bobenko and R. Seiler eds., Clarendon Press, Oxford, 1999, pp. 209–234. MR1676599 (2001b:37101)
- [31] F.W. Nijhoff, V.G. Papageorgiou, H.W. Capel, G.R.W. Quispel, The lattice Gel'fand–Dikii hierarchy, *Inverse Problems*. 8(4), (1992) 597–621. MR1178232 (93h:58077)
- [32] V. Papageorgiou, B. Grammaticos, A. Ramani, Integrable lattices and convergence acceleration algorithms, *Phys. Letters A*, 179 (1993) 111–115. MR1230809 (94b:65009)
- [33] V. Papageorgiou, B. Grammaticos, A. Ramani, Integrable difference equations and numerical analysis algorithms, in *Symmetries and Integrability of Difference Equations*, D. Levi et al. eds., CRM Proceedings and Lecture Notes, vol. 9, AMS, Providence, 1996, pp. 269–279. MR1416845 (97k:39013)
- [34] A. Salam, *Extrapolation: Extension et Nouveaux Résultats*, Thèse, Université des Sciences et Technologies de Lille, 1993.
- [35] A. Salam, On a generalization of the ε -algorithm, *J. Comput. Appl. Math.*, 46 (1993) 455–464. MR1223011 (94a:65002)
- [36] C. Schneider, Vereinfachte Rekursionen zur Richardson–Extrapolation in Spezialfällen, *Numer. Math.*, 24 (1975) 177–184. MR0378342 (51:14510)
- [37] D. Shanks, An analogy between transient and mathematical sequences and some nonlinear sequence-to-sequence transforms suggested by it. Part I, Memorandum 9994, Naval Ordnance Laboratory, White Oak, July 1949.
- [38] D. Shanks, Non linear transformations of divergent and slowly convergent sequences, *J. Math. Phys.*, 34 (1955) 1–42. MR0068901 (16:961e)
- [39] A. Sidi, *Practical Extrapolation Methods. Theory and Applications*, Cambridge University Press, Cambridge, 2003. MR1994507 (2004e:65005)
- [40] W.W. Symes, The QR algorithm and scattering for the nonperiodic Toda lattice, *Physica 4D* (1982) 275–280. MR653781 (83h:58053)
- [41] S. Tsujimoto, Y. Nakamura, M. Iwasaki, The discrete Lotka–Volterra system computes singular values, *Inverse Problems* 17 (2001) 53–58. MR1818491 (2001m:35307)

- [42] R. Vein, P. Dale, *Determinants and Their Applications in Mathematical Physics*, Springer-Verlag, New York, 1999. MR1654469 (99m:15005)
- [43] G. Walz, *Asymptotics and Extrapolation*, Akademie Verlag, Berlin, 1996. MR1386191 (97j:65001)
- [44] E.J. Weniger, Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series, *Comp. Phys. Reports*, 10 (1989) 189–371.
- [45] J. Wimp, *Sequence Transformations and Their Applications*, Academic Press, New York, 1981. MR615250 (84e:65005)
- [46] P. Wynn, On a device for computing the $e_m(S_n)$ transformation, *MTAC*, 10 (1956) 91–96. MR0084056 (18:801e)
- [47] P. Wynn, On a procrustean technique for the numerical transformation of slowly convergent sequences and series, *Proc. Cambridge Phil. Soc.*, 52 (1956) 663–671. MR0081979 (18:478c)
- [48] P. Wynn, Acceleration techniques in numerical analysis, with particular references to problems in one independent variable, in *Proc. IFIP Congress 62, Munich, 27 Aug.-1 Sept. 1962*, C.M. Poplewell ed., North-Holland, Amsterdam, 1962, pp. 149–156.
- [49] P. Wynn, An arsenal of Algol procedures for the evaluation of continued fractions and for effecting the epsilon algorithm, *Chiffres*, 4 (1966) 327–362. MR0203963 (34:3810)
- [50] P. Wynn, Upon systems of recursions which obtain among the quotients of the Padé table, *Numer. Math.*, 8 (1966) 264–269. MR0215499 (35:6339)

LABORATOIRE PAUL PAINLEVÉ, UMR CNRS 8524, UFR DE MATHÉMATIQUES PURES ET APPLIQUÉES, UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE, FRANCE

E-mail address: Claude.Brezinski@univ-lille1.fr

LSEC, INSTITUTE OF COMPUTATIONAL MATHEMATICS AND SCIENTIFIC ENGINEERING COMPUTING, AMSS, CHINESE ACADEMY OF SCIENCES, AND GRADUATE SCHOOL OF THE CHINESE ACADEMY OF SCIENCES, BEIJING, PEOPLE'S REPUBLIC OF CHINA

E-mail address: heyi@lsec.cc.ac.cn

LSEC, INSTITUTE OF COMPUTATIONAL MATHEMATICS AND SCIENTIFIC ENGINEERING COMPUTING, AMSS, CHINESE ACADEMY OF SCIENCES, BEIJING, PEOPLE'S REPUBLIC OF CHINA

E-mail address: hxb@lsec.cc.ac.cn

UNIVERSITÀ DEGLI STUDI DI PADOVA, DIPARTIMENTO DI MATEMATICA PURA ED APPLICATA, ITALY

E-mail address: Michela.RedivoZaglia@unipd.it

LSEC, INSTITUTE OF COMPUTATIONAL MATHEMATICS AND SCIENTIFIC ENGINEERING COMPUTING, AMSS, CHINESE ACADEMY OF SCIENCES, AND GRADUATE SCHOOL OF THE CHINESE ACADEMY OF SCIENCES, BEIJING, PEOPLE'S REPUBLIC OF CHINA

E-mail address: [sunjq@lsec.cc.ac.cn](mailto:sunj@lsec.cc.ac.cn)