Friendly dedicated to Professor Philippe Ciarlet for his 80th birthday

HIROTA'S BILINEAR METHOD, SHANKS' TRANSFORMATION, AND THE ε–ALGORITHMS

C. BREZINSKI and M. REDIVO–ZAGLIA

Communicated by Vasile Brînzănescu

Hirota's bilinear method can be quite useful in the solution of nonlinear differential and difference equations. In this paper, we show how this method can lead to a novel proof that the ε–algorithm of Wynn implements the Shanks' sequence transformation and, reciprocally, that the quantities it computes are expressed as ratios of Hankel determinants as given by Shanks. New identities between Hankel determinants and the quantities involved in Hirota's method are obtained, and they form the basis of our proof. Then, the same bunch of results is showed to hold also for the confluent form of the ε–algorithm. This treatment could also be useful for other sequence transformations and the corresponding recursive algorithms.

AMS 2010 Subject Classification: 65B05, 65B99, 65Q10, 65Q30.

Key words: extrapolation, acceleration methods, sequence transformations, Shanks' transformation, epsilon-algorithm, Hirota's bilinear method.

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 n and k, and converging, under certain assumptions, faster to the same limit, that

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

A well-known example of such a transformation is the Richardson's extrapolation process, which gives rise to the Romberg's method for accelerating the convergence of the trapezoidal rule for approximating a definite integral.

Similarly, let f be a function such that $\lim_{t\to\infty} f(t) = S$. If f tends slowly to S, it can be transformed, by a *function transformation*, into a set of new functions $\{T_k\}$ which, under some assumptions, converge to S faster than f, that is

$$\forall k, \lim_{t\to\infty} \frac{T_k(t)-S}{f(t)-S} = 0 \quad \text{or} \quad \forall t, \lim_{k\to\infty} \frac{T_k(t)-S}{f(t)-S} = 0 \quad \text{or both.}$$

In most of the sequence (and function) transformations, the terms of the new sequences (and the new functions) 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 [16, 31, 33, 34].

Hirota's bilinear method [22] was conceived for resolving integrable nonlinear partial differential or difference evolution equations that have soliton solutions. The aim of this paper is to apply this method to a well known sequence transformation for accelerating the convergence of some sequences due to Shanks [30] which can be implemented via the ε–algorithm of Wynn [35]. Hirota's method was already applied to a multistep generalization of the ε–algorithm [14]. However, since the derivation of this algorithm and the corresponding transformation were quite tedious, the interest of Hirota's method was hidden by the technical difficulties of the proofs. Thus, it seemed to us that it could be interesting to show its interest in a simpler case.

The Shanks' transformation and the ε–algorithm are presented in Section 2. Relations between Hankel determinants are given in Section 3. Hirota's bilinear method is explained in Section 4. These two Sections contain new identities. In Section 5, we first show how Hirota's bilinear method leads to a proof that the ε–algorithm of Wynn implements the Shanks' sequence transformation, and, vice versa, that the quantities computed by this algorithm are expressed by the ratios of Hankel determinants defining the Shanks' transformation. Then, in Section 6, we point out that the same bunch of results also holds for the confluent form of the ε–algorithm [36]. A conclusion with trails for new research ends the paper.

2. THE SHANKS' TRANSFORMATION AND THE ε–ALGORITHM

A well-known sequence transformation is due to Shanks [30]. It consists in transforming (S_n) into a set of sequences $\{(e_k(S_n))\}$ where

$$e_k(S_n) = \frac{H_{k+1}(S_n)}{H_k(\Delta^2 S_n)}, \quad k, n = 0, 1, \ldots, \tag{1}$$

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 $H_k(u_n)$ denotes the following Hankel determinant

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

with $H_0(u_n) = 1$.

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

$$(3) \quad H_k(u_n) = \begin{vmatrix} u_n & \cdots & u_{n+k-1} \\ \Delta u_n & \cdots & \Delta u_{n+k-1} \\ \vdots & & \vdots \\ \Delta^{k-1} u_n & \cdots & \Delta^{k-1} u_{n+k-1} \end{vmatrix} = \begin{vmatrix} u_n & \cdots & \Delta^{k-1} u_n \\ \Delta u_n & \cdots & \Delta^k u_n \\ \vdots & & \vdots \\ \Delta^{k-1} u_n & \cdots & \Delta^{2k-2} u_n \end{vmatrix}.$$

The ε–algorithm is a recursive algorithm due to Wynn [35] for implementing the Shanks' transformation without computing the Hankel determinants appearing in (1). Its rule is

$$(4) \qquad \varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}}, \quad k, n = 0, 1, \ldots$$

with $\varepsilon_{-1}^{(n)} = 0$ and $\varepsilon_0^{(n)} = S_n$, $n = 0, 1, \ldots$, and it holds, for all k and n,

$$(5) \qquad \varepsilon_{2k}^{(n)} = e_k(S_n) \quad \text{and} \quad \varepsilon_{2k+1}^{(n)} = \frac{1}{e_k(\Delta S_n)}.$$

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

$$(6) \qquad \varepsilon_{2k}^{(n)} = \frac{H_{k+1}(S_n)}{H_k(\Delta^2 S_n)} \quad \text{and} \quad \varepsilon_{2k+1}^{(n)} = \frac{H_k(\Delta^3 S_n)}{H_{k+1}(\Delta S_n)}.$$

The proof of these relations was obtained by Wynn by using the Sylvester's determinantal identity (see Section 3) and the Schweins' one, which can both be found, for example, in [1] (see [12, pp. 142–143] for their proofs). The difficulty lays in the nonlinearity of the algorithm. Of course, Wynn's great merit was to discover the rules of the ε–algorithm.

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 what we

will do in this paper, but by a new and different path from that followed by Wynn for deriving his ε–algorithm.

2. *Derivation.* Only the transformation is known, and one has to *derive* the algorithm for its implementation. This was the way followed by Wynn to obtain his ε–algorithm. It was a fruitful idea since it opened the route for obtaining other algorithms (see [18]). This is the case of the E–algorithm which is the most general convergence acceleration algorithm known so far [10]. The topological ε–algorithms were also derived similarly [7, 17].
3. *Formulation.* Only the algorithm is known, and one has to guess a *formula* (for example, a ratio of determinants) for the transformation it implements, and to prove it. This was the situation for the second generalization of the ε–algorithm proposed in [6], whose form was obtained by Salam [27, 28]. The θ–algorithm is also such a recursive algorithm [4], but the formula for the corresponding transformation is still unknown.

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. [13, 23–26]. They are interested by the fact that convergence acceleration algorithms are nonlinear difference equations in two variables whose solutions are explicitly known. An important procedure for obtaining a closed–form solution of soliton equations is Hirota's bilinear method [22] which consists in writing the solution as a ratio, and then working with its numerator and its denominator. In this paper, inspired by the approach followed in [19], we will show how to link the Shanks' transformation and the ε–algorithm by means of Hirota's method. Let us begin by some relations between Hankel determinants that will be needed for that purpose, and some others which could intervene in other transformations and in Padé approximation.

3. RELATIONS BETWEEN HANKEL DETERMINANTS

In this section, recurrence relations between Hankel determinants will be given (see [14] for some of them and their generalization).

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

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

The *Sylvester's determinantal identity* is

$$\begin{aligned}|M|\cdot|A| &= \begin{vmatrix}\alpha & a^T\\ b & A\end{vmatrix}\cdot\begin{vmatrix}A & c\\ d^T & \delta\end{vmatrix}-\begin{vmatrix}a^T & \beta\\ A & c\end{vmatrix}\cdot\begin{vmatrix}b & A\\ \gamma & d^T\end{vmatrix}\\ &= D_1\cdot D_2 - D_3\cdot D_4.\end{aligned} \tag{7}$$

Applying the Sylvester's identity to the Hankel determinant $H_{k+1}(u_n)$ defined by (2) leads to the well-known recurrence relation for these determinants

$$H_{k+1}(u_n)H_{k-1}(u_{n+2}) = H_k(u_n)H_k(u_{n+2}) - [H_k(u_{n+1})]^2. \tag{8}$$

Shanks himself used this relation with $u_n = S_n$ and $u_n = \Delta^2 S_n$ for computing recursively the numerators and the denominators of his transformation (found in 1949 [29], but only published in 1955 [30]).

For $u_n = \Delta^i S_n$ for $i = 0,\ldots,3$, the identity (8) leads to the following known relations which are useful, for example, for simplifying the proofs of the results about the ε–algorithm when applied to totally monotonic and totally oscillating sequences [5,8]. We have

$$\begin{aligned}\varepsilon_{2k}^{(n)} &= \varepsilon_{2k-2}^{(n)} - [H_k(\Delta S_n)]^2/[H_k(\Delta^2 S_n)H_{k-1}(\Delta^2 S_n)]\\ \varepsilon_{2k-1}^{(n)} &= \varepsilon_{2k+1}^{(n)} - [H_k(\Delta^2 S_n)]^2/[H_k(\Delta S_n)H_{k+1}(\Delta S_n)]\\ 1/\varepsilon_{2k+1}^{(n)} &= 1/\varepsilon_{2k-1}^{(n)} - [H_k(\Delta^2 S_n)]^2/[H_k(\Delta^3 S_n)H_{k+1}(\Delta^3 S_n)].\end{aligned}$$

We now consider the determinant

$$|M_1| = \begin{vmatrix}1 & 1 & \cdots & 1 & 1\\ \Delta S_n & \Delta S_{n+1} & \cdots & \Delta S_{n+k} & \Delta S_{n+k+1}\\ \vdots & \vdots & & \vdots & \vdots\\ \Delta^k S_n & \Delta^k S_{n+1} & \cdots & \Delta^k S_{n+k} & \Delta^k S_{n+k+1}\\ S_n & S_{n+1} & \cdots & S_{n+k} & S_{n+k+1}\end{vmatrix} \tag{9}$$

where M_1 is a $(k+2)\times(k+2)$ matrix. Replacing each column, from the last one, by its difference with the previous one, we obtain a determinant whose first row only contains 0 except in the first column where the first 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 $|M_1| = (-1)^k H_{k+1}(\Delta S_n)$. Let us now apply the Sylvester's identity to the matrix M_1. By using the notation given in (7), we see that $|A|$, that is the determinant of the matrix obtained from M_1 by suppressing the first and last rows and columns, is equal to $H_k(\Delta S_{n+1})$, that $D_2 = (-1)^k H_{k+1}(S_{n+1})$, and $D_4 = (-1)^k H_{k+1}(S_n)$. By performing similar manipulations directly on $|M_1|$, we see also that $D_1 = H_k(\Delta^2 S_n)$, and $D_3 = H_k(\Delta^2 S_{n+1})$. We finally obtain

$$H_{k+1}(\Delta S_n)H_k(\Delta S_{n+1}) = H_k(\Delta^2 S_n)H_{k+1}(S_{n+1}) - H_k(\Delta^2 S_{n+1})H_{k+1}(S_n). \tag{10}$$

Directly from (10), by replacing the S_n's by ΔS_n (that is by increasing by 1 all the powers of the operator Δ), we find

(11)
$$H_{k+1}(\Delta^2 S_n)H_k(\Delta^2 S_{n+1}) = H_k(\Delta^3 S_n)H_{k+1}(\Delta S_{n+1}) - H_k(\Delta^3 S_{n+1})H_{k+1}(\Delta S_n).$$

We now apply the Sylvester's determinantal identity to $H_{k+1}(S_n) = (-1)^k D_4$. We directly get

(12)
$$H_{k+1}(S_n)H_{k-1}(\Delta S_{n+1}) = H_k(S_n)H_k(\Delta S_{n+1}) - H_k(S_{n+1})H_k(\Delta S_n).$$

By replacing, as above, the S_n's in (12) by ΔS_n, we obtain

(13)
$$H_{k+1}(\Delta S_n)H_{k-1}(\Delta^2 S_{n+1}) = H_k(\Delta S_n)H_k(\Delta^2 S_{n+1}) - H_k(\Delta S_{n+1})H_k(\Delta^2 S_n).$$

We consider now the following determinant of the $(k+1)\times(k+1)$ matrix M_2

$$|M_2| = \begin{vmatrix} 1 & 1 & \cdots & 1 & 1 \\ \Delta S_n & \Delta S_{n+1} & \cdots & \Delta S_{n+k-1} & \Delta S_{n+k} \\ \vdots & \vdots & & \vdots & \vdots \\ \Delta S_{n+k-2} & \Delta S_{n+k-1} & \cdots & \Delta S_{n+2k-3} & \Delta S_{n+2k-2} \\ \Delta S_{n+k-1} & \Delta S_{n+k} & \cdots & \Delta S_{n+2k-2} & \Delta S_{n+2k-1} \end{vmatrix}.$$

Replacing each column, from the last one, by its difference with the previous one, we obtain again a determinant whose first row only contains 0 except in the first column where the first element is equal to 1. Expanding this determinant with respect to its first row, we have $|M_2| = H_k(\Delta^2 S_n)$.

Applying now the Sylvester's identity to the matrix M_2 we see that $|A| = H_{k-1}(\Delta S_{n+1})$, $D_2 = H_k(\Delta S_{n+1})$, $D_4 = H_k(\Delta S_n)$, and that D_1 and D_3 have the same form as $|M_2|$, without the last row and column for D_1, and without the last row and the first column for D_3. Thus $D_1 = H_{k-1}(\Delta^2 S_n)$ and $D_3 = H_k(\Delta^2 S_{n+1})$, and we finally obtain the relation

(14)
$$H_k(\Delta^2 S_n)H_{k-1}(\Delta S_{n+1}) = H_{k-1}(\Delta^2 S_n)H_k(\Delta S_{n+1}) - H_{k-1}(\Delta^2 S_{n+1})H_k(\Delta S_n).$$

Replacing the S_n's by ΔS_n, we also have

(15)
$$H_k(\Delta^3 S_n)H_{k-1}(\Delta^2 S_{n+1}) = H_{k-1}(\Delta^3 S_n)H_k(\Delta^2 S_{n+1}) - H_{k-1}(\Delta^3 S_{n+1})H_k(\Delta^2 S_n).$$

All the relations given above are, of course, valid independently of the dimension of the Hankel determinants.

Let us now derive a relation for the product of Hankel determinants of the same dimension. We multiply (14) by $H_k(S_{n+1})$, we multiply (12) by $H_{k-1}(\Delta^2 S_{n+1})$, and we subtract. It gives

$$H_{k-1}(\Delta S_{n+1})[H_k(\Delta^2 S_n)H_k(S_{n+1}) - H_{k+1}(S_n)H_{k-1}(\Delta^2 S_{n+1})]$$

$$= H_k(\Delta S_{n+1})[H_k(S_{n+1})H_{k-1}(\Delta^2 S_n) - H_{k-1}(\Delta^2 S_{n+1})H_k(S_n)].$$

Using (10), but with the dimension of the Hankel determinants decreased by 1, we see that the bracket in the right hand side is equal to $H_k(\Delta S_n)H_{k-1}(\Delta S_{n+1})$, and, after simplifying both sides by $H_{k-1}(\Delta S_{n+1})$, we obtain

$$H_k(\Delta S_{n+1})H_k(\Delta S_n) = H_k(\Delta^2 S_n)H_k(S_{n+1}) - H_{k+1}(S_n)H_{k-1}(\Delta^2 S_{n+1}). \tag{16}$$

Of course, the following relation also holds

$$H_k(\Delta^2 S_{n+1})H_k(\Delta^2 S_n) = H_k(\Delta^3 S_n)H_k(\Delta S_{n+1}) - H_{k+1}(\Delta S_n)H_{k-1}(\Delta^3 S_{n+1}). \tag{17}$$

For finding the identities given in this section, the difficulty laid in identifying the determinant to which the Sylvester's determinantal identity had to be applied.

4. THE HIROTA'S BILINEAR METHOD

Hirota's bilinear method [22] is a technique which could be much useful for solving certain nonlinear partial differential and difference equations. It consists in expressing the unknown as a ratio and, then, treating separately the numerator and the denominator.

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

$$\varepsilon_k^{(n)} = \frac{G_k^n}{F_k^n}. \tag{18}$$

Thus, from (6), we have

$$G_{2k}^n = H_{k+1}(S_n), \quad F_{2k}^n = H_k(\Delta^2 S_n), \quad G_{2k+1}^n = H_k(\Delta^3 S_n), \quad F_{2k+1}^n = H_{k+1}(\Delta S_n). \tag{19}$$

These expressions are proved in Sections 5.1 and 5.2. For that purpose, we follow a procedure based on Hirota's bilinear method which is similar to that used in the case of the multistep ε–algorithm [14]. The same method was also adopted in [19] for deriving a determinantal expression for a new convergence acceleration algorithm derived from the lattice Boussinesq equation. The only differences between [19] and what is presented below is that these authors employed the Jacobi's determinantal identity, while we consider the Sylvester's one (which is in fact the same after a permutation of rows and columns), and we do not use the Schweins' identity (since the ε–algorithm is simpler than the algorithm considered in [19]).

Plugging the expressions (19) into the relations (10) to (17) leads respectively to

$$F_{2k+1}^n F_{2k-1}^{n+1} = F_{2k}^n G_{2k}^{n+1} - F_{2k}^{n+1} G_{2k}^n, \tag{20}$$

$$(21) \qquad F_{2k+2}^{n} F_{2k}^{n+1} = G_{2k+1}^{n} F_{2k+1}^{n+1} - G_{2k+1}^{n+1} F_{2k+1}^{n},$$
$$(22) \qquad G_{2k}^{n} F_{2k-3}^{n+1} = G_{2k-2}^{n} F_{2k-1}^{n+1} - G_{2k-2}^{n+1} F_{2k-1}^{n},$$
$$(23) \qquad F_{2k+1}^{n} F_{2k-2}^{n+1} = F_{2k-1}^{n} F_{2k}^{n+1} - F_{2k-1}^{n+1} F_{2k}^{n},$$
$$(24) \qquad F_{2k}^{n} F_{2k-3}^{n+1} = F_{2k-2}^{n} F_{2k-1}^{n+1} - F_{2k-2}^{n+1} F_{2k-1}^{n},$$
$$(25) \qquad G_{2k+1}^{n} F_{2k-2}^{n+1} = G_{2k-1}^{n} F_{2k}^{n+1} - G_{2k-1}^{n+1} F_{2k}^{n},$$
$$(26) \qquad F_{2k-1}^{n+1} F_{2k-1}^{n} = F_{2k}^{n} G_{2k-2}^{n+1} - G_{2k}^{n} F_{2k-2}^{n+1},$$
$$(27) \qquad F_{2k}^{n+1} F_{2k}^{n} = G_{2k+1}^{n} F_{2k-1}^{n+1} - F_{2k+1}^{n} G_{2k-1}^{n+1}.$$

All these relations can be coupled by pair, and each of them can be recovered by replacing k by $2k$ and by $2k+1$ (or by $2k-1$) in the following identities. Thus, the relations (20) and (21) can be gathered in

$$(28) \qquad F_{k+2}^{n} F_{k}^{n+1} = (-1)^{k} [G_{k+1}^{n} F_{k+1}^{n+1} - G_{k+1}^{n+1} F_{k+1}^{n}].$$

Similarly, (22) and (25) cluster into

$$(29) \qquad G_{k}^{n} F_{k-3}^{n+1} = G_{k-2}^{n} F_{k-1}^{n+1} - G_{k-2}^{n+1} F_{k-1}^{n},$$

and, from (23) and (24), we have

$$(30) \qquad F_{k}^{n} F_{k-3}^{n+1} = F_{k-2}^{n} F_{k-1}^{n+1} - F_{k-2}^{n+1} F_{k-1}^{n},$$

and, finally, (26) and (27) can be coupled into

$$(31) \qquad F_{k}^{n+1} F_{k}^{n} = (-1)^{k} [G_{k+1}^{n} F_{k-1}^{n+1} - F_{k+1}^{n} G_{k-1}^{n+1}].$$

Let us give some additional identities which can easily be obtained. Using the relation (8) with $u_n = \Delta^i S_n$ for $i = 0$ and 2, and the expressions (19), gives

$$G_{2k}^{n} G_{2k-4}^{n+2} = G_{2k-2}^{n} G_{2k-2}^{n+2} - [G_{2k-2}^{n+1}]^2$$
$$F_{2k}^{n} F_{2k-4}^{n+2} = F_{2k-2}^{n} F_{2k-2}^{n+2} - [F_{2k-2}^{n+1}]^2.$$

Similarly, when $u_n = \Delta^i S_n$ for $i = 1$ and 3, (8) leads to

$$G_{2k+1}^{n} G_{2k-3}^{n+2} = G_{2k-1}^{n} G_{2k-1}^{n+2} - [G_{2k-1}^{n+1}]^2$$
$$F_{2k+1}^{n} F_{2k-3}^{n+2} = F_{2k-1}^{n} F_{2k-1}^{n+2} - [F_{2k-1}^{n+1}]^2.$$

These four identities can be gathered into two, valid for k even or odd

$$G_{k}^{n} G_{k-4}^{n+2} = G_{k-2}^{n} G_{k-2}^{n+2} - [G_{k-2}^{n+1}]^2$$
$$F_{k}^{n} F_{k-4}^{n+2} = F_{k-2}^{n} F_{k-2}^{n+2} - [F_{k-2}^{n+1}]^2.$$

Finally, applying the Sylvester's identity to the second determinant in (3) with $u_n = S_n$ and $u_n = \Delta S_n$ leads to

$$G_{2k}^{n} F_{2k-2}^{n} = G_{2k-2}^{n} F_{2k}^{n} - [F_{2k-1}^{n}]^2$$

$$F^n_{2k+1}G^n_{2k-1} \quad = \quad F^n_{2k-1}G^n_{2k+1} - [F^n_{2k}]^2,$$

which can be coupled into

$$[F^n_{k-1}]^2 = (-1)^k[G^n_{k-2}F^n_k - G^n_kF^n_{k-2}].$$

5. HIROTA'S METHOD, THE ε–ALGORITM, AND THE SHANKS' TRANFORMATION

In this section, we will show how the relations obtained from Hirota's method allow to recover the Shanks' transformation from the recursive rule of the ε–algorithm and, then, how this rule can be deduced from the definition of the transformation.

5.1. FROM THE ε–ALGORITHM TO THE SHANKS' TRANSFORMATION

In this section, starting from the rule (4) of the ε–algorithm, we will show how to recover the determinantal expressions (5) and (6) linking it with the Shanks' transformation.

Plugging (18) into the recursive rule (4) of the ε–algorithm, we get

$$\frac{G^n_{k+1}}{F^n_{k+1}} - \frac{G^{n+1}_{k-1}}{F^{n+1}_{k-1}} = \frac{1}{\dfrac{G^{n+1}_k}{F^{n+1}_k} - \dfrac{G^n_k}{F^n_k}}$$

that is

$$\frac{G^n_{k+1}F^{n+1}_{k-1} - F^n_{k+1}G^{n+1}_{k-1}}{F^n_{k+1}F^{n+1}_{k-1}} = \frac{F^{n+1}_kF^n_k}{G^{n+1}_kF^n_k - F^{n+1}_kG^n_k}. \tag{32}$$

Equating the numerators and equating the denominators in both sides of this identity, we obtain the coupled relations

$$G^n_{k+1}F^{n+1}_{k-1} - F^n_{k+1}G^{n+1}_{k-1} \quad = \quad (-1)^kF^{n+1}_kF^n_k \tag{33}$$

$$G^{n+1}_kF^n_k - F^{n+1}_kG^n_k \quad = \quad (-1)^kF^n_{k+1}F^{n+1}_{k-1}. \tag{34}$$

Although it does not appear in (32) since it could be cancelled out, the sign $(-1)^k$ in these relations is needed to recover (31) and (28). By (19), these relations are those given in Section 3 among the Hankel determinants which show that the ratios of determinants defining the Shanks' transformation have been recovered.

Remark 1. Let us remind that, as noticed in [2] and fully explained in [11], the $\varepsilon_k^{(n)}$'s can be written as

$$\varepsilon_{2k}^{(n)} = \frac{f_k(S_n, \ldots, S_{n+2k})}{Df_k(S_n, \ldots, S_{n+2k})}, \qquad \varepsilon_{2k+1}^{(n)} = \frac{Df_k(\Delta S_n, \ldots, \Delta S_{n+2k})}{f_k(\Delta S_n, \ldots, \Delta S_{n+2k})},$$

where f_k is a function depending on $2k+1$ variables and such that $D^2 f_k \equiv 0$, where Df_k denotes the sum of the partial derivatives of f_k. We thus obtain the following connection with Hirota's bilinear method

$$\begin{array}{ll} G_{2k}^n = f_k(S_n, \ldots, S_{n+2k}), & F_{2k}^n = Df_k(S_n, \ldots, S_{n+2k}), \\ G_{2k+1}^n = Df_k(\Delta S_n, \ldots, \Delta S_{n+2k}), & F_{2k+1}^n = f_k(\Delta S_n, \ldots, \Delta S_{n+2k}). \end{array}$$

5.2. FROM THE SHANKS' TRANSFORMATION TO THE ε–ALGORITHM

Starting from the definition of Shanks' transformation as $e_k(S_n) = H_{k+1}(S_n)/H_k(\Delta^2 S_n)$, we will prove that these quantities, also denoted by $\varepsilon_{2k}^{(n)}$, can be recursively computed by the rule of the ε–algorithm (4) where $\varepsilon_{2k+1}^{(n)} = 1/e_k(\Delta S_n)$, and that these $\varepsilon_{2k+1}^{(n)}$'s can be themselves computed by the same recursive rule.

The proof can be conducted in two different ways. The first one consists in using directly the relations between the Hankel determinants proved in Section 3, while the second considers the corresponding identities among the F_k^n's and the G_k^n's given in Section 4 together with their definitions (19).

We choose the second procedure because the coupled relations given above allow to treat simultaneously the cases where k is odd or even, which is not the case with the Hankel determinants. Of course, by using the definitions (19) in (18), and replacing k by $2k$, we have

$$\varepsilon_{2k}^{(n)} = G_{2k}^n / F_{2k}^n = H_{k+1}(S_n)/H_k(\Delta^2 S_n) = e_k(S_n),$$

and replacing k by $2k+1$

$$\varepsilon_{2k+1}^{(n)} = G_{2k+1}^n / F_{2k+1}^n = H_k(\Delta^3 S_n)/H_{k+1}(\Delta S_n) = 1/e_k(\Delta S_n).$$

Writing (28) with the index $k-1$ instead of k and using (31), we obtain by division

$$\frac{F_k^{n+1} F_k^n}{(-1)^{k-1}[G_k^n F_k^{n+1} - G_k^{n+1} F_k^n]} = 1 = \frac{(-1)^k [G_{k+1}^n F_{k-1}^{n+1} - F_{k+1}^n G_{k-1}^{n+1}]}{F_{k+1}^n F_{k-1}^{n+1}}.$$

Dividing by $F_k^{n+1}F_k^n$ the numerator and the denominator of the first ratio, and simplifying the second ratio leads to

$$\frac{1}{G_k^n/F_k^n - G_k^{n+1}/F_k^{n+1}} = \frac{G_{k-1}^{n+1}}{F_{k-1}^{n+1}} - \frac{G_{k+1}^n}{F_{k+1}^n},$$

which is nothing else than the rule (4) of the ε–algorithm

$$1/(\varepsilon_k^{(n)} - \varepsilon_k^{(n+1)}) = \varepsilon_{k-1}^{(n+1)} - \varepsilon_{k+1}^{(n)}.$$

Since the $\varepsilon_{2k+1}^{(n)}$'s are intermediate results, they can be eliminated thus leading to a relation between quantities with an even lower index. But, similarly, the $\varepsilon_{2k}^{(n)}$ can be eliminated and a relation between quantities with an odd lower index is obtained. This is the so–called *cross rule* carried out by Wynn [37] directly from the rule of the ε–algorithm

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

The validity of this rule can be verified by our approach. Using (18) and the relationships (28), (30), and (31), we arrive at a tautology which proves the rule.

6. THE CONFLUENT ε–ALGORITHM

The *confluent form of the ε–algorithm* was obtained by Wynn [36] by replacing, in the rule (4) of his scalar algorithm, the discrete variable n by the continuous one $t + nh$, $\varepsilon_{2k+1}^{(n)}$ by $\varepsilon_{2k+1}(t)/h$, $\varepsilon_{2k}^{(n)}$ by $\varepsilon_{2k}(t)$, and, then, letting h tend to 0. Thus, he obtained

$$\varepsilon_{k+1}(t) = \varepsilon_{k-1}(t) + \frac{1}{\varepsilon_k'(t)}, \tag{35}$$

with $\varepsilon_{-1}(t) = 0$ and $\varepsilon_0(t) = f(t)$. It holds

$$\varepsilon_{2k}(t) = \frac{H_{k+1}^{(0)}(t)}{H_k^{(2)}(t)} \quad \text{and} \quad \varepsilon_{2k+1}(t) = \frac{H_k^{(3)}(t)}{H_{k+1}^{(1)}(t)}, \tag{36}$$

where $H_k^{(n)}(t)$ denotes the following funtional Hankel determinant

$$H_k^{(n)}(t) = \begin{vmatrix} f^{(n)}(t) & f^{(n+1)}(t) & \cdots & f^{(n+k-1)}(t) \\ f^{(n+1)}(t) & f^{(n+2)}(t) & \cdots & f^{(n+k)}(t) \\ \vdots & \vdots & & \vdots \\ f^{(n+k-1)}(t) & f^{(n+k)}(t) & \cdots & f^{(n+2k-2)}(t) \end{vmatrix}, \tag{37}$$

with $\forall n, H_0^{(n)}(t) = 1$ and where $f^{(i)}$ is the i–th derivative of the function f.

This algorithm is aimed at transforming f into a set of functions $\{\varepsilon_{2k}\}$ which, under some assumptions, converge to S, the limit of $f(t)$ when t tends to infinity, faster than f. With notations similar to those used for the Shanks' transformation, we set $\varepsilon_{2k}(t) = e_k(f)(t)$, then $\varepsilon_{2k+1}(t) = 1/e_k(f')(t)$.

We will now show that Hirota's method can be an alternative to Wynn's approach for linking the confluent Shanks' transformation and the corresponding confluent form of the ε–algorithm. This technique was already employed in [32] for treating the confluent form of the discrete algorithm presented in [19].

Setting

$$\varepsilon_k(t) = \frac{G_k(t)}{F_k(t)},$$

we have from (36)

$$G_{2k}(t) = H_{k+1}^{(0)}(t), \quad F_{2k}(t) = H_k^{(2)}(t), \quad G_{2k+1}(t) = H_k^{(3)}(t), \quad F_{2k+1}(t) = H_{k+1}^{(1)}(t). \tag{38}$$

Then, applying Hirota's bilinear method to the recursive rule (35) of the confluent form of the ε–algorithm, leads to the coupled equations

$$G_{k+1}(t)F_{k-1}(t) - F_{k+1}(t)G_{k-1}(t) = (-1)^k F_k^2(t) \tag{39}$$

$$G_k'(t)F_k(t) - G_k(t)F_k'(t) = (-1)^k F_{k+1}(t)F_{k-1}(t). \tag{40}$$

For proving that the F_k's and G_k's can be expressed by formulæ (38), we have to prove that determinantal identities similar to those given in Section 3 hold for the functional Hankel determinants (37). We consider the second determinant in (3) with $u_n = \Delta^i S_n$

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

Replacing, in this determinant, $\Delta^j S_n$ by $f^{(j)}(t+nh)$ for all j, dividing the terms of the first column by h^i, those of the second column by h^{i+1}, and so on, and then performing the same operations on the rows from the first one, and finally letting h tend to zero, we get $H_k^{(i)}(t)$ (see [36] for detailed explanations).

Then, all the determinantal identities of Section 3, previously defined by applying the Sylvester's determinantal identity to determinants of the form of (3) and (2), with the appropriate changes, are valid for the functional Hankel determinants. In particular, when h tends to 0, (39) directly follows from (33), and (40) from (34).

Finally, all the results given in the preceding Sections for the scalar ε–algorithm are also valid for its confluent form (see [15] for a generalization to the multistep ε–algorithm).

7. CONCLUSION AND FUTURE RESEARCH

In this paper, after giving identities relating Hankel determinants and explaining Hirota's bilinear method, we were first able to *show* that the quantities $\varepsilon_k^{(n)}$ computed by the ε–algorithm are expressed as ratios of Hankel determinants related to the Shanks' transformation, and conversely to *derive* the recursive rule (4) of the ε–algorithm from the determinantal formulæ defining the Shanks' transformation. It must be noticed that, contrarily to the approach of Wynn [35], we do not make use of the Schweins' determinantal identity. The difficult point in these tasks was to find the determinants to which the Sylvester's determinantal identity had to be applied. Our approach was then extended to the confluent case.

The approach developed in this paper could possibly be extended to other nonlinear convergence acceleration algorithms such as, for example, the two generalizations of the ε–algorithm given in [6], or the one proposed in [21], or its q-difference version [20], or the ρ–algorithm. Other algorithms related to them, such as the qd, the η, the ω, and the rs–algorithms, and the g–decomposition, could also be treated in a similar way (see [16] for their definitions). The quantities computed by these algorithms can all be represented by ratios of determinants. These extensions, as well as extensions to other acceleration algorithms such as the θ–algorithm [4], will be the subject of future research.

Acknowledgements. The work of C.B. was supported in part by the Labex CEMPI (ANR-11-LABX-0007-01). M.R.-Z. is member of the INdAM Research group GNCS. Her work was partially supported by the University of Padua, Project No. DOR1807909/18.

REFERENCES

[1] A.C. Aitken, *Determinants and Matrices.* Oliver and Boyd, Edinburgh and London, 1949.

[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, *Convergence d'une forme confluente de l'ε–algorithme.* C. R. Acad. Sci. Paris, Sér A **273** (1971), 582–585.

[4] C. Brezinski, *Accélération de suites à convergence logarithmique.* C. R. Acad. Sci. Paris, Sér. A **273** (1971), 727–730.

[5] C. Brezinski, *Méthodes d'Accélération de la Convergence en Analyse Numérique.* Thèse de Doctorat d'État ès Sciences Mathématiques, Université Scientifique et Médicale de Grenoble, Grenoble, 26 avril 1971.

[6] C. Brezinski, *Conditions d'application et de convergence de procédés d'extrapolation.* Numer. Math. **20** (1972), 64–79.

[7] C. Brezinski, *Généralisation de la transformation de Shanks, de la table de Padé et de l'ε–algorithme.* Calcolo **12** (1975), 317–360.

[8] C. Brezinski, *Accélération de la Convergence en Analyse Numérique.* Lecture Notes in Math. **584**, Springer-Verlag, Heidelberg, 1977.

[9] C. Brezinski, *Sur le calcul de certains rapports de déterminants.* In: L. Wuytack (Ed.), *Padé Approximation and its Applications.* Lecture Notes in Math. **765**, Springer–Verlag, Heidelberg, 1979, pp. 184–210.

[10] C. Brezinski, *A general extrapolation algorithm.* Numer. Math. **35** (1980), 175–187.

[11] C. Brezinski, *Quasi-linear extrapolation processes.* In: R.P. Agarwal *et al.* (Eds.), *Numerical Mathematics. Singapore 1988.* ISNM **86**, Birkhäuser, Basel, 1988, pp. 61–78.

[12] C. Brezinski, *Biorthogonality and its Applications of Numerical Analysis.* Marcel Dekker, New York, 1992.

[13] C. Brezinski, *Cross rules and non-Abelian lattice equations for the discrete and confluent non–scalar epsilon-algorithms.* J. Phys. A **43** (2010), 205–201.

[14] C. Brezinski, Y. He, X.-B. Hu, M. Redivo–Zaglia and J.-Q. Sun, *Multistep epsilon-algorithm, Shanks' transformation, and the Lotka-Volterra system by Hirota's method.* Math. Comp. **81** (2012), 1527–1549.

[15] C. Brezinski, Y. He, X.-B. Hu, J.-Q. Sun and H.-W. Tam, *Confluent form of the multistep epsilon-algorithm, and the relevant integrable system.* Stud. Appl. Math. **127** (2011), 191–209.

[16] C. Brezinski and M. Redivo-Zaglia, *Extrapolation Methods. Theory and Practice.* North–Holland, Amsterdam, 1991.

[17] C. Brezinski and M. Redivo-Zaglia, *The simplified topological ε–algorithms for accelerating sequences in a vector space.* SIAM J. Sci. Comput. **36** (2014), A2227–A2247.

[18] C. Brezinski and M. Redivo-Zaglia, *The genesis and early developments of Aitken's process, Shanks' transformation, and related fixed point methods.* Numer. Algorithms **80** (2019), doi: 10.1007/s11075-018-0567-2.

[19] Y. He, X.-B. Hu, J.-Q. Sun and E.J. Weniger, *Convergence acceleration algorithm via the lattice Boussinesq equation.* SIAM J. Sci. Comput. **33** (2011), 1234–1245.

[20] Y. He, X.-B. Hu and H.-W. Tam, *Q-difference version of the ε-algorithm.* J. Phys. A **42** (2009), 095202.

[21] Y. He, X.-B. Hu, H.-W. Tam and S. Tsujimoto, *Convergence acceleration algorithms related to a general E-transformation and its particular cases.* Japan J. Indust. Appl. Math. **30** (2013), 263–285.

[22] R. Hirota, *The Direct Method in Soliton Theory.* Cambridge Univ. Press, Cambridge, 1992.

[23] A. Nagai and J. Satsuma, *Discrete soliton equations and convergence acceleration algorithms.* Phys. Lett. A **209** (1995), 305–312.

[24] A. Nagai, T. Tokihiro and J. Satsuma, *The Toda molecule equation and the ε–algorithm.* Math. Comp. **67** (1998), 1565–1575.

[25] V. Papageorgiou, B. Grammaticos and A. Ramani, *Integrable difference equations and numerical analysis algorithms.* In: D. Levi *et al.* (Eds.), *Symmetries and Integrability of Difference Equations.* CRM Proceedings and Lecture Notes **9**, AMS, Providence, 1996, pp. 269–279.

[26] V. Papageorgiou, B. Grammaticos and A. Ramani, *Integrable lattices and convergence acceleration algorithms.* Phys. Lett. A **179** (1993), 111–115.

[27] A. Salam, *Extrapolation: Extension et Nouveaux Résultats.* Thèse, Université des Sciences et Technologies de Lille, 1993.

[28] A. Salam, *On a generalization of the ε-algorithm.* J. Comput. Appl. Math. **46** (1993), 455–464.

[29] 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.

[30] D. Shanks, *Non linear transformations of divergent and slowly convergent sequences.* J. Math. Phys. **34** (1955), 1–42.

[31] A. Sidi, *Practical Extrapolation Methods. Theory and Applications.* Cambridge Univ. Press, Cambridge, 2003.

[32] J.-Q. Sun, Y. He, X.-B. Hu and H.-W. Tam, *Q-difference and confluent forms of the lattice Boussinesq equation and the relevant convergence acceleration algorithms.* J. Math. Phys. **52** (2011), 023522.

[33] E.J. Weniger, *Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series.* Comp. Phys. Reports **10** (1989), 189–371.

[34] J. Wimp, *Sequence Transformations and their Applications.* Academic Press, New York, 1981.

[35] P. Wynn, *On a device for computing the $e_m(S_n)$ transformation.* Math. Tables Aids Comput. **10** (1956), 91–96.

[36] P. Wynn, *Confluent forms of certain nonlinear algorithms.* Arch. Math. **11** (1960), 223–234.

[37] P. Wynn, *Upon systems of recursions which obtain among the quotients of the Padé table.* Numer. Math. **8** (1966), 264–269.

Received 5 April 2018

Université de Lille,
CNRS, UMR 8524 - Laboratoire Paul Painlevé,
F-59000 Lille, France
Claude.Brezinski@univ-lille1.fr

Università degli Studi di Padova,
Dipartimento di Matematica
"Tullio Levi-Civita",
Via Trieste 63,
35121–Padova, Italy
Michela.RedivoZaglia@unipd.it