Multistep ε–algorithm and Shanks' transformation by Hirota's method

Michela Redivo–Zaglia
University of Padua - Italy

Joint work with C. Brezinski, Y. He, X.-B. Hu, J.-Q. Sun

- ➜ Shanks' transformation and the ε–algorithm
- ➜ Relations between Hankel determinants
- ➜ The Hirota's bilinear method
- ➜ From the ε–algorithm to Shanks' transformation
- ➜ From Shanks' transformation to the ε–algorithm
- ➜ The multistep ε–algorithm
- ➜ Extended discrete Lotka–Volterra equation
- ➜ Conclusions

Hirota's bilinear method was invented for resolving integrable nonlinear partial differential or difference evolution equations that have soliton solutions.

We will show how Hirota's bilinear method can lead to a proof that the **ε–algorithm** of Wynn implements **Shanks' sequence transformation** and, reciprocally, that the quantities computed by this algorithm are expressed by the ratios of **Hankel determinants** defining Shanks' transformation.

The link between this algorithm and the **Lotka–Volterra** equation will be studied.

Then we will propose a **multistep extension** of Shanks' transformation, the ε–algorithm, and the Lotka–Volterra equation.

Shanks' transformation and the ε–algorithm

Shanks' transformation **(1949, 1955)** consists in transforming $(\mathbf{S_n})$ into the set of sequences $\{(\mathbf{e_k}(\mathbf{S_n})\}$ given by

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

where Δ is the usual forward difference operator and where $\mathbf{H_k}(\mathbf{u_n})$ denotes the **Hankel determinants**

$$\mathbf{H_k}(\mathbf{u_n}) = \begin{vmatrix} u_n & u_{n+1} & \cdots & u_{n+k-1} \\ \Delta u_n & \Delta u_{n+1} & \cdots & \Delta u_{n+k-1} \\ \vdots & \vdots & & \vdots \\ \Delta^{k-1} u_n & \Delta^{k-1} u_{n+1} & \cdots & \Delta^{k-1} u_{n+k-1} \end{vmatrix}, \quad \mathbf{H_0}(\mathbf{u_n}) = 1.$$

The ε**–algorithm** is a recursive algorithm due to **Wynn (1956)** for implementing Shanks' transformation without computing the Hankel determinants. Its rule is

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

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

It holds, for all k and n,

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

The $\varepsilon_{2k+1}^{(n)}$'s are intermediated results, and it holds

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

Sylvester's determinantal identity

Let $\mathbf{A}$ be a **square matrix**, α, β, γ and δ numbers, $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\mathbf{d}$ **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

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

Wynn proved that Shanks' transformation can be implemented via the ε–algorithm. His proof was based on Sylvester's and Schweins determinantal identities.

Recently, a generalization of Shanks' transformation was proposed by **Hu, Weniger et al. (SISC 2011)**. It can be implemented by a two–step ε–algorithm whose construction relies on **Hirota's bilinear method (1992)**.

These results were further extended by **Brezinski, Hu, R.-Z. et al. (AMS MCOM, in press)** to a multilevel Shanks's transformation and the corresponding multilevel ε–algorithm.

Since this last extension needs a pretty complicated explanation, in order to give a flavour of these ideas, we will show how they can also be used in the case of the original Shanks' transformation and the ε–algorithm of Wynn.

Relations between Hankel determinants

First, we will give some relations between Hankel determinants that will be used in the sequel.
We consider the following determinant

$$D = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \Delta S_n & \Delta S_{n+1} & \cdots & \Delta S_{n+k+1} \\ \vdots & \vdots & & \vdots \\ \Delta^k S_n & \Delta^k S_{n+1} & \cdots & \Delta^k S_{n+k+1} \\ S_n & S_{n+1} & \cdots & S_{n+k+1} \end{vmatrix}.$$

It is easy to show that, after some algebraic manipulations,

$$D = (-1)^k H_{k+1}(\Delta S_n).$$

Applying only the Sylvester's identity to D, in different ways, after some algebraic manipulations, we obtain several relations. For instance

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

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

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

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

$$H_k(\Delta S_{n+1})H_k(\Delta S_n) =$$

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

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

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

$$H_{k-1}(\Delta S_{n+1})[H_k(\Delta^3 S_n)H_k(\Delta S_{n+1}) - H_{k+1}(\Delta S_n)H_{k-1}(\Delta^3 S_{n+1})] =$$

$$H_k(\Delta^2 S_{n+1})[H_k(\Delta S_{n+1})H_{k-1}(\Delta^3 S_n) - H_{k-1}(\Delta^3 S_{n+1})H_k(\Delta S_n)]$$

. . . .

. . . .

The Hirota's bilinear method

Hirota's bilinear method is a technique which could be much useful for solving certain nonlinear differential and difference equations. It consists in expressing the unknown as a ratio of quantities and, then, treating separately the numerator and the denominator, trying to find the quantities for which the equality holds.

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

$$\varepsilon_{\mathbf{k}}^{(\mathbf{n})} = \frac{\mathbf{G}_{\mathbf{k}}^{\mathbf{n}}}{\mathbf{F}_{\mathbf{k}}^{\mathbf{n}}}.$$

Plugging this expression into the recursive rule of the ε–algorithm given by Wynn, we get

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

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

and equating the numerators and the denominators of both sides of this last identity, we obtain the following **coupled relations**

$$\mathbf{G_{k+1}^n F_{k-1}^{n+1}-F_{k+1}^n G_{k-1}^{n+1} = (-1)^k F_k^{n+1} F_k^n} \qquad (1)$$

$$\mathbf{G_k^{n+1} F_k^n-F_k^{n+1} G_k^n = (-1)^k F_{k+1}^n F_{k-1}^{n+1}}. \qquad (2)$$

Link between ε–algorithm and Shanks' transformation

We will now prove that the quantities $\varepsilon_{\mathbf{k}}^{(\mathbf{n})}$ computed by the rule of the ε–algorithm, are expressed by the ratio $\varepsilon_{\mathbf{k}}^{(\mathbf{n})} = \dfrac{\mathbf{G}_{\mathbf{k}}^{\mathbf{n}}}{\mathbf{F}_{\mathbf{k}}^{\mathbf{n}}}$,

when we set

Even lower indexes	**Odd lower indexes**
$\mathbf{G}_{2k}^{n} = \mathbf{H}_{k+1}(\mathbf{S}_{n})$	$\mathbf{G}_{2k+1}^{n} = \mathbf{H}_{k}(\boldsymbol{\Delta}^{3}\mathbf{S}_{n})$
$\mathbf{F}_{2k}^{n} = \mathbf{H}_{k}(\boldsymbol{\Delta}^{2}\mathbf{S}_{n})$	$\mathbf{F}_{2k+1}^{n} = \mathbf{H}_{k+1}(\boldsymbol{\Delta}\mathbf{S}_{n})$

We assume that the preceding determinantal expressions for the $\mathbf{G}_{\mathbf{k}}^{\mathbf{n}}$ and $\mathbf{F}_{\mathbf{k}}^{\mathbf{n}}$ hold true.

We use some of the determinantal relations between Hankel determinants previously given, obtained by using Sylvester determinantal identity.

By replacing the Hankel determinants by the corresponding $\mathbf{G_k^n}$'s and $\mathbf{F_k^n}$'s , we obtain the two relations

$$\begin{aligned} G_{2k+2}^{n}F_{2k}^{n+1} - F_{2k+2}^{n}G_{2k}^{n+1} &= -F_{2k+1}^{n+1}F_{2k+1}^{n} \\ G_{2k+1}^{n}F_{2k-1}^{n+1} - F_{2k+1}^{n}G_{2k-1}^{n+1} &= F_{2k}^{n+1}F_{2k}^{n}, \end{aligned}$$

which gives the coupled relation **(1)** given above, that is

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

when the lower index $\mathbf{k}$ is, respectively, odd and even.

Similarly, we can obtain the following relations

$$\begin{aligned} G^{n+1}_{2k+1}F^{n}_{2k+1} - F^{n+1}_{2k+1}G^{n}_{2k+1} &= -F^{n}_{2k+2}F^{n+1}_{2k} \\ G^{n+1}_{2k}F^{n}_{2k} - F^{n+1}_{2k}G^{n}_{2k} &= F^{n}_{2k+1}F^{n+1}_{2k-1} \end{aligned}$$

that leads to the coupled relation **(2)** given above, that is

$$\mathbf{G^{n+1}_{k}F^{n}_{k} - F^{n+1}_{k}G^{n}_{k} = (-1)^{k}F^{n}_{k+1}F^{n+1}_{k-1}},$$

when the lower index $\mathbf{k}$ is, respectively, odd and even.

Thus, when we assume that the determinantal expressions for the $\mathbf{G^{n}_{k}}$ and $\mathbf{F^{n}_{k}}$ hold, the **ε–algorithm implements the Shanks' transformation**, that is

$$\mathbf{e^{(n)}_{k}(S_n) = \varepsilon^{(n)}_{2k} = \frac{H_{k+1}(S_n)}{H_{k}(\Delta^2 S_n)}}.$$

It is also possible to verify that the determinantal definitions of the Shanks' transformation

$$\mathbf{e_k(S_n) = \frac{H_{k+1}(S_n)}{H_k(\Delta^2 S_n)}} \quad \text{and} \quad \mathbf{\frac{1}{e_k(\Delta S_n)} = \frac{H_k(\Delta^3 S_n)}{H_{k+1}(\Delta S_n)}}$$

and the recursive rule of the ε–algorithm with

$$\mathbf{e_k(S_n) = \varepsilon^{(n)}_{2k+2}} \quad \text{and} \quad \mathbf{\frac{1}{e_k(\Delta S_n)} = \varepsilon^{(n)}_{2k+1}}$$

produce **identical results**.

This verification is based on **tricky manipulations of the identities and the relations given above**.

It's too technical to give them in details.

We were able to show that (rule of the ε–algorithm with **even lower indexes**)

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

since we obtained

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

and

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

Similarly (rule of the ε–algorithm with **odd lower indexes**)

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

since we obtained

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

and

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

The confluent ε-algorithm

Similar results can be obtained through the **Hirota's bilinear method** for the confluent form of the ε–algorithm obtained by **Wynn (1960)** that transforms a function f into a set of functions $\{\varepsilon_{2k}\}$ which, under some assumptions, converge to S, the limit of $f(t)$ when $t \longrightarrow \infty$, faster than f. The rule is

$$\varepsilon_{\mathbf{k+1}}(\mathbf{t}) = \varepsilon_{\mathbf{k-1}}(\mathbf{t}) + \frac{\mathbf{1}}{\varepsilon'_{\mathbf{k}}(\mathbf{t})},$$

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

$$\varepsilon_{\mathbf{2k}}(\mathbf{t}) = \frac{\mathbf{H}^{(\mathbf{0})}_{\mathbf{k+1}}(\mathbf{t})}{\mathbf{H}^{(\mathbf{2})}_{\mathbf{k}}(\mathbf{t})} \quad \text{and} \quad \varepsilon_{\mathbf{2k+1}}(\mathbf{t}) = \frac{\mathbf{H}^{(\mathbf{3})}_{\mathbf{k}}(\mathbf{t})}{\mathbf{H}^{(\mathbf{1})}_{\mathbf{k+1}}(\mathbf{t})},$$

where

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

with $H_0^{(n)}(t) = 1$.

Setting (Hirota's bilinear method)

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

and plugging this expression into the recursive rule of the confluent ε–algorithm, leads to the **coupled equations**

$$G_{k+1}(t)F_{k-1}(t) - F_{k+1}(t)G_{k-1}(t) = (-1)^k F_k^2(t) \tag{3}$$

$$G_k'(t)F_k(t) - G_k(t)F_k'(t) = (-1)^k F_{k+1}(t)F_{k-1}(t), \tag{4}$$

that hold true if

Even lower indexes	**Odd lower indexes**
$G_{2k}(t) = H_{k+1}^{(0)}(t)$	$G_{2k+1}(t) = H_k^{(3)}(t)$
$F_{2k}(t) = H_k^{(2)}(t)$	$F_{2k+1}(t) = H_{k+1}^{(1)}(t)$

The Lotka–Volterra equation

By differentiating the recursive rule of the confluent form of the ε–algorithm, and by setting $\mathbf{M_k(t)} = \varepsilon'_{\mathbf{k}}(\mathbf{t})$, we obtain a **closed–form solution** of the difference-differential equation

$$\mathbf{M'_k(t) = M_k^2(t)[M_{k-1}(t) - M_{k+1}(t)].}$$

If we set $\mathbf{N_k(t) = M_k(t)M_{k+1}(t)}$, this relation becomes

$$M'_k(t) = M_k(t)[N_{k-1}(t) - N_k(t)].$$

Moreover, by replacing the expressions for $\mathbf{M'_k(t)}$ and $\mathbf{M'_{k+1}(t)}$ into

$$N'_k(t) = M'_k(t)M_{k+1}(t) + M_k(t)M'_{k+1}(t),$$

we obtain the **Lotka–Volterra equation**

$$\mathbf{N'_k(t) = N_k(t)[N_{k-1}(t) - N_{k+1}(t)].}$$

Now, since $\mathbf{M_k(t)} = \varepsilon'_{\mathbf{k}}(\mathbf{t})$ and $\varepsilon_{\mathbf{k}}(\mathbf{t}) = \dfrac{\mathbf{G_k(t)}}{\mathbf{F_k(t)}}$, we also have

$$M_k(t) = \frac{G'_k(t)F_k(t) - G_k(t)F'_k(t)}{F_k^2(t)},$$

and it follows

$$N_k(t) = \frac{G'_k(t)F_k(t) - G_k(t)F'_k(t)}{F_k^2(t)} \frac{G'_{k+1}(t)F_{k+1}(t) - G_{k+1}(t)F'_{k+1}(t)}{F_{k+1}^2(t)}.$$

Thanks to **(4)**, we finally obtain a **closed–form solution** of the Lotka–Volterra equation

$$\mathbf{N_k(t) = -\frac{F_{k-1}(t)F_{k+2}(t)}{F_k(t)F_{k+1}(t)}},$$

that is

$$\begin{aligned}\mathbf{N_{2k}(t)} &= -\frac{\mathbf{H_k^{(1)}(t)H_{k+1}^{(2)}(t)}}{\mathbf{H_k^{(2)}(t)H_{k+1}^{(1)}(t)}}\\ \mathbf{N_{2k+1}(t)} &= -\frac{\mathbf{H_k^{(2)}(t)H_{k+2}^{(1)}(t)}}{\mathbf{H_{k+1}^{(1)}(t)H_{k+1}^{(2)}(t)}},\end{aligned}$$

with $\mathbf{N_{-1}(t) = 0}$ and $\mathbf{N_0(t) = -f''(t)/f'(t)}$.

The multistep ε–algorithm

A two-step generalization of Shanks' transformation and the ε–algorithm were obtained by **Hu, Weniger et al.**
We extended this work to the **multistep Shanks transformation**

$$\mathbf{e_{k,m}(S_n)} = \frac{\mathbf{H_{k+1}^{(m)}(S_n)}}{\mathbf{H_k^{(m)}(\Delta^{m+1} S_n)}},$$

where the determinants $\mathbf{H_k^{(m)}}$ (which depend on m) are

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

The multistep Shanks' transformation can be implemented by the **multistep ε–algorithm**

$$\varepsilon^{(n)}_{k+1,m} = \varepsilon^{(n+1)}_{k-m,m} + \frac{1}{\prod_{i=1}^{m}(\varepsilon^{(n+1)}_{k-m+i,m} - \varepsilon^{(n)}_{k-m+i,m})}, \quad k, n = 0, 1, \ldots,$$

with the initial values

$$\varepsilon^{(n)}_{-m,m} = 0,\ \varepsilon^{(n)}_{-m+1,m} = \varepsilon^{(n)}_{-m+2,m} = \cdots = \varepsilon^{(n)}_{-1,m} = n,\ \varepsilon^{(n)}_{0,m} = S_n, \quad n = 0, 1, \ldots.$$

Displaying these quantities in a **double array** similar to the ε–array, we see that this rule relates $2m+2$ quantities located in an extended lozenge covering $m+2$ columns (the first lower index represents the column in this array) and two descending diagonals as showed below

$$
\begin{array}{ccccccc}
 & \varepsilon_{k-m+1,m}^{(n)} & & & & & \\
\varepsilon_{k-m,m}^{(n+1)} & & \varepsilon_{k-m+2,m}^{(n)} & & & & \\
 & \varepsilon_{k-m+1,m}^{(n+1)} & & \ddots & & & \\
 & & \ddots & & \ddots & & \\
 & & & \ddots & & \varepsilon_{k,m}^{(n)} & \\
 & & & & \varepsilon_{k-1,m}^{(n+1)} & & \boldsymbol{\varepsilon_{k+1,m}^{(n)}} \\
 & & & & & \varepsilon_{k,m}^{(n+1)} &
\end{array}
$$

By **Hirota's bilinear method**, following the same lines as for the ε–algorithm for extending the determinantal relations, we are able to prove that

$$\varepsilon^{(n)}_{(m+1)k,m} = e_{k,m}(S_n) = \frac{H^{(m)}_{k+1}(S_n)}{H^{(m)}_k(\Delta^{m+1}S_n)}$$

and that

$$\begin{aligned}
\varepsilon^{(n)}_{(m+1)(k-1)+1,m} &= \frac{H^{(m)}_{k-1}(\Delta^{m+2}S_n)}{H^{(m)}_k(\Delta S_n)}, \\
\varepsilon^{(n)}_{(m+1)(k-1)+i,m} &= \frac{\Phi^{(m)}_{k+1}(\Delta^{i-1}S_n)}{H^{(m)}_k(\Delta^{i}S_n)}, \quad i = 2, 3, \ldots, m,
\end{aligned}$$

where the determinants $\Phi^{(m)}_k$, which also depend on m, are given by

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

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

Sorry, but the proofs are too technical to be given here !!

The multistep Shanks' transformation can be implemented by the E**–algorithm** with the initializations $\mathbf{g_i(n) = \Delta^{im} S_n}$ for $i = 1, 2, \cdots$, and for all n, and we get, for all k and n,

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

Thus, we have the

Theorem 1

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

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

Let us remind that the kernel of the **original Shanks' transformation**

$$\mathbf{e_{km}} : (\mathbf{S_n}) \longmapsto (\mathbf{e_{km}}(\mathbf{S_n}) = \varepsilon_{\mathbf{2km}}^{(\mathbf{n})})$$

is the set of sequences such that, for all n,

$$\mathbf{S_n = S + b_1 \Delta S_n + \cdots + b_{km} \Delta^{km} S_n},$$

where $b_1, \ldots, b_{km}$, $b_{km} \neq 0$, are constants. Thus, we have the

Corollary 1

The kernel of the multistep Shanks' transformation $\mathbf{e_{k,m}}$ is contained into the kernel of the Shanks' transformation $\mathbf{e_{km}}$.

An extended discrete Lotka–Volterra equation

If we set $\left(a^{(n)}_{k-\frac{m-1}{2}}\right)^{-1} = \varepsilon^{(n+1)}_{k,m} - \varepsilon^{(n)}_{k,m}$, then the relation of the multistep ε–algorithm is transformed into the **extended discrete Lotka–Volterra equation**

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

This equation can be considered as the **time discretization**, for $N=-1$, of the **extended Lotka–Volterra equation**

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

Conclusions

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 (already done by Hu et al.), or its two generalizations, or the general ε–algorithm, or the ρ–algorithm, and the γ–algorithm.

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.

References

- C. Brezinski, Y. He, X.-B. Hu, M. Redivo-Zaglia, J.-Q. Sun, *Multistep ε-algorithm, Shanks' transformation, and Lotka-Volterra system by Hirota's method*, Math. Comput., in press.
- 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.
- C. Brezinski, M. Redivo–Zaglia, *Extrapolation Methods. Theory and Practice*, North–Holland, Amsterdam, 1991.
- Y. He, X.–B. Hu, J.–Q. Sun, E.J. Weniger, *Convergence acceleration algorithm via an equation related to the lattice Boussinesq equation*, SIAM J. Sci. Comput, 33 (2011) 1234–1245.

- R. Hirota, *The Direct Method in Soliton Theory*, Cambridge University Press, Cambridge, 1992.
- 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.
- D. Shanks, *Non linear transformations of divergent and slowly convergent sequences*, J. Math. Phys., 34 (1955) 1–42.
- P. Wynn, On a device for computing the $e_m(S_n)$ transformation, MTAC, 10 (1956) 91–96.
- P. Wynn, Confluent forms of certain nonlinear algorithms, Arch. Math., 11 (1960) 223-234.