A general extrapolation procedure revisited

C. Brezinski

Laboratoire d'Analyse Numérique et d'Optimisation, UFR IEEA-M3, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq Cedex, France

E-mail: brezinsk@omega.univ-lille1.fr

M. Redivo-Zaglia

Dipartimento di Elettronica e Informatica, Università degli Studi di Padova, Via Gradenigo 6/a, 35131 Padova, Italy

E-mail: elen@elett1.dei.unipd.it

Received 6 August 1993; revised 30 May 1994

The E-algorithm is the most general extrapolation algorithm actually known. The aim of this paper is to provide a new approach to this algorithm. This approach gives a deeper insight into the E-algorithm, and allows one to obtain new properties and to relate it to other algorithms. Some extensions of the procedure are discussed.

Keywords: Convergence acceleration, extrapolation.

AMS(MOS) subject classification: 65B05.

1. Introduction

The E-algorithm is the most general extrapolation algorithm actually known. It was first obtained by an interpolation procedure by Schneider in 1975 [21] and then by Meinardus and Taylor [14] in the context of best uniform approximation. However, its present formulation as an extrapolation procedure was only given some years later by Håvie [12] and Brezinski [2]. Håvie's approach was by elimination while Brezinski used Sylvester's determinantal identity. The quantities computed by the E-algorithm can be expressed as ratios of determinants and they also satisfy a recursive scheme which generalizes the Richardson extrapolation process. All these approaches are reviewed and explained in [4]. The E-algorithm contains most of the extrapolation algorithms found in the literature (see [7] for a review) and it has interesting acceleration properties [2, 13, 20, 24].

The aim of this paper is to give a new derivation of the E-algorithm. This derivation is based on the concepts of the annihilation difference operator and remainder estimate introduced by Weniger [25] and developed by Brezinski and Matos [6].

© J.C. Baltzer AG, Science Publishers

This new approach also leads us to new results about the E-algorithm and it clarifies its properties. Some extensions will be discussed.

2. The mechanism of extrapolation

Let $u = (u_n)$ be an arbitrary sequence. We shall denote by 1 the sequence whose terms are all equal to 1. We shall define the sequence (N_k) of difference operators by

$$N_0^{(n)}(u) = u_n,$$

$$N_{k+1}^{(n)}(u) = \Delta\left(\frac{N_k^{(n)}(u)}{N_k^{(n)}(g_{k+1})}\right),$$

where $N_k^{(n)}(u)$ denotes the nth term of the sequence $(N_k(u))$ and where $g_i = (g_i(n))$ are given auxiliary sequence for $i = 1, 2, \ldots$.

The sequence $S = (S_n)$ to be transformed is assumed to satisfy, $\forall n$,

$$S_\infty - S_n = a_1 g_1(n) + a_2 g_2(n) + \cdots .$$

This relation can be written as

$$S_\infty N_0^{(n)}(1) - N_0^{(n)}(S) = a_1 N_0^{(n)}(g_1) + a_2 N_0^{(n)}(g_2) + \cdots.$$

We have

$$S_\infty \frac{N_0^{(n)}(1)}{N_0^{(n)}(g_1)} - \frac{N_0^{(n)}(S)}{N_0^{(n)}(g_1)} = a_1 + a_2 \frac{N_0^{(n)}(g_2)}{N_0^{(n)}(g_1)} + a_3 \frac{N_0^{(n)}(g_3)}{N_0^{(n)}(g_1)} + \cdots.$$

Applying the operator Δ to both sides we have, since $\Delta a_1 = 0$,

$$S_\infty \Delta\left(\frac{N_0^{(n)}(1)}{N_0^{(n)}(g_1)}\right) - \Delta\left(\frac{N_0^{(n)}(S)}{N_0^{(n)}(g_1)}\right) = a_2 \Delta\left(\frac{N_0^{(n)}(g_2)}{N_0^{(n)}(g_1)}\right) + \cdots,$$

that is, using the definition of N_1,

$$S_\infty N_1^{(n)}(1) - N_1^{(n)}(S) = a_2 N_1^{(n)}(g_2) + a_3 N_1^{(n)}(g_3) + \cdots.$$

Obviously, the process can be repeated and we obtain for $k = 0, 1, \ldots$

$$S_\infty N_k^{(n)}(1) - N_k^{(n)}(S) = a_{k+1} N_k^{(n)}(g_{k+1}) + a_{k+2} N_k^{(n)}(g_{k+2}) + \cdots.$$

Dividing both sides by $N_k^{(n)}(g_{k+1})$, we have

$$S_\infty \frac{N_k^{(n)}(1)}{N_k^{(n)}(g_{k+1})} - \frac{N_k^{(n)}(S)}{N_k^{(n)}(g_{k+1})} = a_{k+1} + a_{k+2} \frac{N_k^{(n)}(g_{k+2})}{N_k^{(n)}(g_{k+1})} + \cdots.$$

Applying Δ and since $\Delta a_{k+1} = 0$, we obtain

$$S_\infty \Delta\left(\frac{N_k^{(n)}(1)}{N_k^{(n)}(g_{k+1})}\right) - \Delta\left(\frac{N_k^{(n)}(S)}{N_k^{(n)}(g_{k+1})}\right) = a_{k+2}\Delta\left(\frac{N_k^{(n)}(g_{k+2})}{N_k^{(n)}(g_{k+1})}\right) + \cdots,$$

that is, by definition of N_{k+1}

$$S_\infty N_{k+1}^{(n)}(1) - N_{k+1}^{(n)}(S) = a_{k+2}N_{k+1}^{(n)}(g_{k+2}) + a_{k+3}N_{k+1}^{(n)}(g_{k+3}) + \cdots,$$

and so on.

For any arbitrary sequence $u = (u_n)$, we shall define the sequence transformation E_k by

$$E_k : u = (u_n) \mapsto E_k(u) = \left(E_k^{(n)}(u)\right),$$

where

$$E_k^{(n)}(u) = \frac{N_k^{(n)}(u)}{N_k^{(n)}(1)}.$$

By construction, the quantities $E_k^{(n)}(S)$ are the same as the quantities $E_k^{(n)}$ obtained by the E-algorithm as we shall see in section 3.

Dividing both sides of the relation given above for an arbitrary value of k by $N_k^{(n)}(1)$ and making use of the definition of $E_k^{(n)}(u)$, we obtain

$$S_\infty - E_k^{(n)}(S) = a_{k+1}E_k^{(n)}(g_{k+1}) + a_{k+2}E_k^{(n)}(g_{k+2}) + \cdots,$$

which is a known property of the E-algorithm.

The preceding definition of $E_k^{(n)}(u)$ makes use of $N_k^{(n)}(u)$. The converse also holds and we have:

PROPERTY 1

$$N_k^{(n)}(u) = \Delta\left(\frac{E_{k-1}^{(n)}(u)}{E_{k-1}^{(n)}(g_k)}\right).$$

Proof

We have

$$N_k^{(n)}(u) = \Delta\left(\frac{N_{k-1}^{(n)}(u)}{N_{k-1}^{(n)}(g_k)}\right) = \Delta\left(\frac{N_{k-1}^{(n)}(u)}{N_{k-1}^{(n)}(1)}\,\frac{N_{k-1}^{(n)}(1)}{N_{k-1}^{(n)}(g_k)}\right)$$

and the result follows from the definition of $E_k^{(n)}(u)$. □

Let us now give some more properties of the difference operators N_k and the transformations E_k.

PROPERTY 2

$$\forall k \geq 1, N_k^{(n)}(g_i) = E_k^{(n)}(g_i) = 0 \text{ for } i = 1, \ldots, k.$$

Proof

By induction. We have

$$N_1^{(n)}(g_1) = \Delta\left(\frac{N_0^{(n)}(g_1)}{N_0^{(n)}(g_1)}\right) = \Delta(1) = 0.$$

Thus, the property is true for $k = 1$. Let us assume that it is true for k.

We have, for $i = 1, \ldots, k$,

$$N_{k+1}^{(n)}(g_i) = \Delta\left(N_k^{(n)}(g_i)/N_k^{(n)}(g_{k+1})\right) = \Delta(0) = 0.$$

For $i = k + 1$, we have

$$N_{k+1}^{(n)}(g_{k+1}) = \Delta\left(N_k^{(n)}(g_{k+1})/N_k^{(n)}(g_{k+1})\right) = \Delta(1) = 0.$$

Thus, the property is true for any k.

It follows that, $\forall k \geq 1$,

$$E_k^{(n)}(g_i) = 0$$

for $i = 1, \ldots, k$. □

We have

$$N_{k+1}^{(n)}(u) = \frac{N_k^{(n+1)}(u)}{N_k^{(n+1)}(g_{k+1})} - \frac{N_k^{(n)}(u)}{N_k^{(n)}(g_{k+1})},$$

which shows that the $N_k^{(n)}(u)$'s satisfy a triangular recursive scheme. Thus, following the theory developed by Brezinski and Walz [9], these quantities can be expressed as a ratio of two determinants. Moreover, using the property above and theorem 3.2 of Brezinski and Walz, we have:

PROPERTY 3

$$N_k^{(n)}(u) = \frac{\begin{vmatrix} u_n & \cdots & u_{n+k} \\ g_1(n) & \cdots & g_1(n+k) \\ \vdots & & \vdots \\ g_k(n) & \cdots & g_k(n+k) \end{vmatrix}}{\begin{vmatrix} g_0(n) & \cdots & g_0(n+k) \\ g_1(n) & \cdots & g_1(n+k) \\ \vdots & & \vdots \\ g_k(n) & \cdots & g_k(n+k) \end{vmatrix}} N_k^{(n)}(g_0),$$

where $g_0 = (g_0(n))$ is an arbitrary sequence such that $\forall k, n, N_k^{(n)}(g_0) \neq 0$.

The usual determinantal expression for $E_k^{(n)}(u)$ immediately follows from this representation:

PROPERTY 4

$$E_k^{(n)}(u) = \begin{vmatrix} u_n & \cdots & u_{n+k} \\ g_1(n) & \cdots & g_1(n+k) \\ \vdots & & \vdots \\ g_k(n) & \cdots & g_k(n+k) \end{vmatrix} \Big/ \begin{vmatrix} 1 & \cdots & 1 \\ g_1(n) & \cdots & g_1(n+k) \\ \vdots & & \vdots \\ g_k(n) & \cdots & g_k(n+k) \end{vmatrix}.$$

Denoting a determinant by its first column, we have, from Sylvester's determinantal identity

$$\begin{aligned} &|u_n g_1(n) \cdots g_k(n) g_{k+1}(n)| \, |g_1(n+1) \cdots g_k(n+1)| \\ &\quad = |u_n g_1(n) \cdots g_k(n)| \, |g_1(n+1) \cdots g_{k+1}(n+1)| \\ &\quad - |u_{n+1} g_1(n+1) \cdots g_k(n+1)| \, |g_1(n) \cdots g_{k+1}(n)|, \end{aligned}$$

that is,

$$\begin{aligned} &\frac{|u_n g_1(n) \cdots g_{k+1}(n)| \, |g_1(n+1) \cdots g_k(n+1)|}{|g_1(n) \cdots g_{k+1}(n)| \, |g_1(n+1) \cdots g_{k+1}(n+1)|} \\ &\quad = \frac{|u_n g_1(n) \cdots g_k(n)|}{|g_1(n) \cdots g_{k+1}(n)|} - \frac{|u_{n+1} g_1(n+1) \cdots g_k(n+1)|}{|g_1(n+1) \cdots g_k(n+1)|} \\ &\quad = (-1)^{k+1} \Delta \left(\frac{N_k^{(n)}(u)}{N_k^{(n)}(g_{k+1})} \right) = (-1)^{k+1} N_{k+1}^{(n)}(u). \end{aligned}$$

Thus we finally obtain:

PROPERTY 5

$$N_{k+1}^{(n)}(u) = (-1)^{k+1} \frac{|u_n g_1(n) \cdots g_{k+1}(n)|\,|g_1(n+1) \cdots g_k(n+1)|}{|g_1(n) \cdots g_{k+1}(n)|\,|g_1(n+1) \cdots g_{k+1}(n+1)|},$$

with the convention that $|g_1(n+1) \ldots g_k(n+1)| = 1$ if $k = 0$.

This property shows that, in fact, the definition of the difference operators N_k is equivalent to Sylvester's determinantal identity.

PROPERTY 6

$$N_k^{(n)}(u) = \sum_{i=0}^{k} B_i^{(k,n)} u_{n+i},$$

with

$$B_0^{(0,n)} = 1$$

and

$$B_0^{(k+1,n)} = -B_0^{(k,n)} / N_k^{(n)}(g_{k+1}),$$

$$B_i^{(k+1,n)} = \frac{B_{i-1}^{(k,n+1)}}{N_k^{(n+1)}(g_{k+1})} - \frac{B_i^{(k,n)}}{N_k^{(n)}(g_{k+1})}, \quad i = 1, \ldots, k,$$

$$B_{k+1}^{(k+1,n)} = \frac{B_k^{(k,n+1)}}{N_k^{(n+1)}(g_{k+1})}.$$

Proof

Obvious from the definition of $N_{k+1}^{(n)}(u)$. □

From this property, we immediately obtain:

PROPERTY 7

$$E_k^{(n)}(u) = \sum_{i=0}^{k} A_i^{(k,n)} u_{n+i},$$

with

$$A_i^{(k,n)} = B_i^{(k,n)} / \sum_{j=0}^{k} B_j^{(k,n)}.$$

This property was already given in [2] and [12].

From the properties 3 and 4, we have:

PROPERTY 8

$$N_k^{(n)}(\Delta u) = N_k^{(n)}(Eu) - N_k^{(n)}(u),$$

$$E_k^{(n)}(\Delta u) = E_k^{(n)}(Eu) - E_k^{(n)}(u),$$

where $Eu = (u_{n+1})$ and $\Delta u = (\Delta u_n)$.

We shall see now how the E-algorithm can be recovered from the above recurrence relations.

3. The E-algorithm

We have

$$E_k^{(n)}(u) = \frac{N_k^{(n)}(u)}{N_k^{(n)}(1)} = \frac{\Delta\left(\dfrac{N_{k-1}^{(n)}(u)}{N_{k-1}^{(n)}(g_k)}\right)}{\Delta\left(\dfrac{N_{k-1}^{(n)}(1)}{N_{k-1}^{(n)}(g_k)}\right)}$$

$$= \frac{\Delta\left(\dfrac{N_{k-1}^{(n)}(u)}{N_{k-1}^{(n)}(1)} \dfrac{N_{k-1}^{(n)}(1)}{N_{k-1}^{(n)}(g_k)}\right)}{\Delta\left(\dfrac{N_{k-1}^{(n)}(1)}{N_{k-1}^{(n)}(g_k)}\right)}$$

$$= \frac{\Delta(E_{k-1}^{(n)}(u)/E_{k-1}^{(n)}(g_k))}{\Delta(1/E_{k-1}^{(n)}(g_k))}$$

$$= \frac{E_{k-1}^{(n+1)}(g_k)E_{k-1}^{(n)}(u) - E_{k-1}^{(n)}(g_k)E_{k-1}^{(n+1)}(u)}{E_{k-1}^{(n+1)}(g_k) - E_{k-1}^{(n)}(g_k)},$$

which is called the rule of the E-algorithm. It may also be expressed as

$$\frac{E_k^{(n)}(u) - E_{k-1}^{(n+1)}(u)}{\Delta E_{k-1}^{(n)}(u)} = \frac{E_{k-1}^{(n+1)}(g_k)}{\Delta E_{k-1}^{(n)}(g_k)}.$$

Thus, setting

$$E_k^{(n)} = E_k^{(n)}(S),$$

$$g_{k,i}^{(n)} = E_k^{(n)}(g_i),$$

we obtain

$$E_k^{(n)} = \frac{g_{k-1,k}^{(n+1)} E_{k-1}^{(n)} - g_{k-1,k}^{(n)} E_{k-1}^{(n+1)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}},$$

$$g_{k,i}^{(n)} = \frac{g_{k-1,k}^{(n+1)} g_{k-1,i}^{(n)} - g_{k-1,k}^{(n)} g_{k-1,i}^{(n+1)}}{g_{k-1,k}^{(n+1)} - g_{k-1,k}^{(n)}},$$

with $E_0^{(n)} = S_n$ and $g_{0,i}^{(n)} = g_i(n)$, which is the E-algorithm.

In the case $g_i(n) = x_n^{1-i}$, where (x_n) is an auxiliary known sequence, the W-algorithm of Sidi [23] is recovered; see also [7].

The formulae of the E-algorithm are one-step formulae allowing one to compute the $E_{k+1}^{(n)}(u)$'s from the $E_k^{(n)}(u)$'s. It is also possible to use multistep formulae allowing one to compute directly the $E_{k+m}^{(n)}(u)$'s from the $E_m^{(n)}(u)$'s without computing the intermediate steps. Such formulae were given in [3]. We shall see now how they can be recovered from our mechanism. Let m be a fixed non-negative integer. We define the operators $\hat{N}_k$ by

$$\widehat{N_{k+1}^{(n)}}(u) = \Delta\left(\frac{\widehat{N_k^{(n)}}(u)}{\widehat{N_k^{(n)}}(\hat{g}_{k+1})}\right),$$

with $\widehat{N_0^{(n)}}(u) = N_m^{(n)}(u)$ and $\hat{g}_i(n) = g_{m+i}(n)$.

We define also the transformation $\hat{E}_k$ by

$$\widehat{E_k^{(n)}}(u) = \widehat{N_k^{(n)}}(u)/\widehat{N_k^{(n)}}(1).$$

We have:

PROPERTY 9

$$\widehat{N_k^{(n)}}(u) = N_{k+m}^{(n)}(u), \quad \widehat{E_k^{(n)}}(u) = E_{k+m}^{(n)}(u).$$

Proof

The second relation obviously follows from the first one which is proved by induction. The property is true for $k = 0$. Let us assume that it is true for k. We have

$$\widehat{N_{k+1}^{(n)}}(u) = \Delta\left(\frac{N_{k+m}^{(n)}(u)}{N_{k+m}^{(n)}(g_{m+k+1})}\right) = N_{m+k+1}^{(n)}(u). \qquad \Box$$

By using property 4, a determinantal formula expressing the quantities $E_{k+m}^{(n)}(u)$ in terms of the $E_m^{(n)}(u)$'s can be obtained. Such a formula is useful when a division by zero (or by a quantity close to zero) occurs in the E-algorithm. It allows one to jump over the singularity (or the near-singularity) and to avoid a breakdown or a near-breakdown in the algorithm; see [5].

Let us consider a variant of the E-algorithm where, now, we set

$$E_k^{(n)}(u) = \frac{N_k^{(n)}(u)}{N_k^{(n)}(v)},$$

where v is an arbitrary sequence such that the denominator in the preceding expression does not vanish. Then, the same relations as above hold after replacing the sequence 1 by the sequence v. Thus, the E-algorithm can still be used for computing recursively these quantities $E_k^{(n)}(u)$ but, now, with the initializations

$$E_0^{(n)} = E_0^{(n)}(S) = S_n/v_n \quad \text{and} \quad g_{0,i}^{(n)} = E_0^{(n)}(g_i) = g_i(n)/v_n.$$

4. The Ford–Sidi algorithm

Let us set

$$\psi_k^{(n)}(u) = N_k^{(n)}(u)/N_k^{(n)}(g_{k+1}).$$

Thus

$$N_{k+1}^{(n)}(u) = \Delta\psi_k^{(n)}(u) = \psi_k^{(n+1)}(u) - \psi_k^{(n)}(u),$$

$$N_{k+1}^{(n)}(g_{k+2}) = \Delta\psi_k^{(n)}(g_{k+2}) = \psi_k^{(n+1)}(g_{k+2}) - \psi_k^{(n)}(g_{k+2}),$$

$$N_{k+1}^{(n)}(1) = \Delta\psi_k^{(n)}(1) = \psi_k^{(n+1)}(1) - \psi_k^{(n)}(1),$$

and it follows that

$$\psi_{k+1}^{(n)}(u) = \Delta\psi_k^{(n)}(u)/\Delta\psi_k^{(n)}(g_{k+2}).$$

We have

$$E_{k+1}^{(n)}(u) = \frac{N_{k+1}^{(n)}(u)}{N_{k+1}^{(n)}(1)} = \frac{\psi_{k+1}^{(n)}(u)\Delta\psi_k^{(n)}(g_{k+2})}{\psi_{k+1}^{(n)}(1)\Delta\psi_k^{(n)}(g_{k+2})} = \frac{\psi_{k+1}^{(n)}(u)}{\psi_{k+1}^{(n)}(1)}$$

and we recover the algorithm proposed by Ford and Sidi [11] for implementing the E-transformation.

The definition of $E_k^{(n)}(u)$ does not make use of g_{k+1}, while the computation of $E_k^{(n)}(u)$ by the Ford–Sidi algorithm does. This is an important drawback of this procedure which, on the other hand, requires less arithmetical operations than the E-algorithm. However, it must be noticed that, if $E_{k+1}^{(n)}(u)$ is not to be computed, then the auxiliary sequence g_{k+1} used in the computation of $\psi_k^{(n)}(u)$ can be arbitrarily chosen.

It must be noticed that

$$\psi_k^{(n)}(u) = \frac{N_k^{(n)}(u)}{N_k^{(n)}(1)} \frac{N_k^{(n)}(1)}{N_k^{(n)}(g_{k+1})} = \frac{E_k^{(n)}(u)}{E_k^{(n)}(g_{k+1})}$$

and thus

$$\psi_k^{(n)}(S) = E_k^{(n)} / g_{k,k+1}^{(n)},$$

$$\psi_k^{(n)}(1) = 1/g_{k,k+1}^{(n)},$$

$$\psi_k^{(n)}(g_{k+1}) = 1.$$

In fact, the quantities $\psi_k^{(n)}(u)$ are exactly the generalized divided differences introduced by Mühlbach [15, 16] and the previous algorithm is exactly the recurrence relationship they satisfy [17–19, 3].

5. The general ε-algorithm

In [10], Carstensen proposed a general ε-algorithm including the ε-algorithm of Wynn [26] which is used for implementing Shanks' sequence transformation [22]. We shall see now, how this general ε-algorithm can be recovered and we shall also give some new properties for it.

We set

$$e_{2k}^n = E_k^{(n)}(S),$$

$$e_{2k+1}^n = \frac{\Delta E_k^{(n)}(\alpha)}{\Delta E_k^{(n)}(S)},$$

$$\mu_0^n = \Delta \alpha_n,$$

and

$$\mu_k^n = \left(e_{2k}^n - e_{2k-2}^{n+1}\right)\left(e_{2k-1}^{n+1} - e_{2k-1}^n\right),$$

where $\alpha = (\alpha_n)$ is an arbitrary sequence such that the quantities involving α in a denominator do not vanish.

Replacing the quantities e_k^n by their definitions and using the rule of the E-algorithm we immediately obtain:

PROPERTY 10

$$\mu_k^n = -\Delta E_{k-1}^{(n)}(S) \frac{E_{k-1}^{(n+1)}(g_k)}{\Delta E_{k-1}^{(n)}(g_k)} \Delta\left(\frac{\Delta E_{k-1}^{(n)}(\alpha)}{\Delta E_{k-1}^{(n)}(S)}\right).$$

In order to recover the general ε-algorithm of Carstensen, we also have to prove that

$$\mu_k^n = \left(e_{2k+1}^n - e_{2k-1}^{n+1}\right)\left(e_{2k}^{n+1} - e_{2k}^n\right).$$

We compose the identity

$$\Delta E_k^{(n)}(\alpha) = \Delta E_{k-1}^{(n+1)}(\alpha) + \Delta\left(\Delta E_{k-1}^{(n)}(\alpha)\frac{E_k^{(n)}(\alpha) - E_{k-1}^{(n+1)}(\alpha)}{\Delta E_{k-1}^{(n)}(\alpha)}\right).$$

But, from the rule of the E-algorithm, the quantity $(E_k^{(n)}(u) - E_{k-1}^{(n+1)}(u))/\Delta E_{k-1}^{(n)}(u)$ is equal to $E_{k-1}^{(n+1)}(g_k)/\Delta E_{k-1}^{(n)}(g_k)$, which shows that it is independent of the sequence u. Thus, writing this quantity with the sequence S instead of α, we have

$$\Delta E_k^{(n)}(\alpha) = \Delta E_{k-1}^{(n+1)}(\alpha) + \Delta\left(\Delta E_{k-1}^{(n)}(\alpha)\frac{E_k^{(n)}(S) - E_{k-1}^{(n+1)}(S)}{\Delta E_{k-1}^{(n)}(S)}\right)$$

$$= \Delta E_{k-1}^{(n+1)}(\alpha) + \frac{\Delta E_{k-1}^{(n+1)}(\alpha)}{\Delta E_{k-1}^{(n+1)}(S)}\Delta\left(E_k^{(n)}(S) - E_{k-1}^{(n+1)}(S)\right)$$

$$+ \Delta\left(\frac{\Delta E_{k-1}^{(n)}(\alpha)}{\Delta E_{k-1}^{(n)}(S)}\right)\left(E_k^{(n)}(S) - E_{k-1}^{(n+1)}(S)\right).$$

Multiplying and dividing the first term in the right hand side by $\Delta E_{k-1}^{(n+1)}(S)$ and using the above expressions together with the rule of the E-algorithm, we obtain

$$\Delta E_k^{(n)}(\alpha) = e_{2k-1}^{n+1}\Delta E_k^{(n)}(S) + \mu_k^n,$$

which is the second relation of Carstensen.

Thus, we finally obtain Carstensen's algorithm

$$e_{2k}^n = e_{2k-2}^{n+1} + \frac{\mu_k^n}{e_{2k-1}^{n+1} - e_{2k-1}^n},$$

$$e_{2k+1}^n = e_{2k-1}^{n+1} + \frac{\mu_k^n}{e_{2k}^{n+1} - e_{2k}^n},$$

with μ_k^n as given in the property above. Let us remark that Carstensen defined μ_k^n as a ratio of a product of determinants and thus property 10 provides a new expression for it.

Let us now prove some new results about this algorithm. We set

$$\tilde{N}_0^{(n)}(u) = u_n,$$

$$\tilde{N}_{k+1}^{(n)}(u) = \Delta\left(\frac{\tilde{N}_k^{(n)}(u)}{\tilde{N}_k^{(n)}(\Delta g_{k+1})}\right),$$

$$\tilde{E}_k^{(n)}(u) = \frac{\tilde{N}_k^{(n)}(u)}{\tilde{N}_k^{(n)}(1)}.$$

$(\tilde{E}_k^{(n)}(u))$ is the sequence obtained by applying the E-algorithm to the sequence u but with the (Δg_i)'s as auxiliary sequences instead of the g_i's.

We have:

PROPERTY 11

$$\frac{\Delta E_k^{(n)}(\alpha)}{\Delta E_k^{(n)}(\beta)} = \frac{\tilde{E}_k^{(n)}(\Delta\alpha)}{\tilde{E}_k^{(n)}(\Delta\beta)},$$

where α and β are two arbitrary sequences.

Proof

By induction. For $k = 0$, we have

$$\frac{\Delta E_0^{(n)}(\alpha)}{\Delta E_0^{(n)}(\beta)} = \frac{\Delta\alpha_n}{\Delta\beta_n} = \frac{E_0^{(n)}(\Delta\alpha)}{E_0^{(n)}(\Delta\beta)},$$

which shows that the property is true for $k = 0$.

Let us assume that the property is true up to the index k. We have

$$\tilde{N}_{k+1}^{(n)}(\Delta u) = \Delta\left(\frac{\tilde{N}_k^{(n)}(\Delta u)}{\tilde{N}_k^{(n)}(\Delta g_{k+1})}\right)$$

$$= \Delta\left(\frac{\tilde{E}_k^{(n)}(\Delta u)}{\tilde{E}_k^{(n)}(\Delta g_{k+1})}\right)$$

$$= \Delta\left(\frac{\Delta E_k^{(n)}(u)}{\Delta E_k^{(n)}(g_{k+1})}\right)$$

by the induction assumption.

Thus

$$\frac{\tilde{E}_{k+1}^{(n)}(\Delta\alpha)}{\tilde{E}_{k+1}^{(n)}(\Delta\beta)} = \frac{\Delta E_k^{(n+1)}(\alpha)\Delta E_k^{(n)}(g_{k+1}) - \Delta E_k^{(n)}(\alpha)\Delta E_k^{(n+1)}(g_{k+1})}{\Delta E_k^{(n+1)}(\beta)\Delta E_k^{(n)}(g_{k+1}) - \Delta E_k^{(n)}(\beta)\Delta E_k^{(n+1)}(g_{k+1})}.$$

By the rule of the E-algorithm

$$E_{k+1}^{(n)}(u) = E_k^{(n+1)}(u) - \frac{\Delta E_k^{(n)}(u)}{\Delta E_k^{(n)}(g_{k+1})} E_k^{(n+1)}(g_{k+1})$$

and

$$E_{k+1}^{(n+1)}(u) = E_k^{(n+1)}(u) - \frac{\Delta E_k^{(n+1)}(u)}{\Delta E_k^{(n+1)}(g_{k+1})} E_k^{(n+1)}(g_{k+1}).$$

Thus

$$\Delta E_{k+1}^{(n)}(u) = \frac{E_k^{(n+1)}(g_{k+1})}{\Delta E_k^{(n)}(g_{k+1})\Delta E_k^{(n+1)}(g_{k+1})}(\Delta E_k^{(n)}(u)\Delta E_k^{(n+1)}(g_{k+1})$$

$$-\Delta E_k^{(n+1)}(u)\Delta E_k^{(n)}(g_{k+1}))$$

and the result follows. □

This property is a generalization of property 10 of [5]. It can be written as:

PROPERTY 12

$$\frac{\Delta E_k^{(n)}(\alpha)}{\tilde{E}_k^{(n)}(\Delta\alpha)}$$

is independent of the sequence α.

Proof

From the preceding property we have

$$\frac{\Delta E_k^{(n)}(\alpha)}{\tilde{E}_k^{(n)}(\Delta\alpha)} = \frac{\Delta E_k^{(n)}(\beta)}{\tilde{E}_k^{(n)}(\Delta\beta)},$$

which proves the result. □

Using these results, we have

$$e_{2k+1}^n = \frac{\tilde{E}_k^{(n)}(\Delta\alpha)}{\tilde{E}_k^{(n)}(\Delta S)}.$$

From the determinantal expressions given above, this is exactly the definition of e_{2k+1}^n given by Carstensen. We also have

$$\mu_k^n = \frac{\tilde{E}_{k-1}^{(n)}(\Delta\alpha)\tilde{E}_{k-1}^{(n+1)}(\Delta S) - \tilde{E}_{k-1}^{(n)}(\Delta S)\tilde{E}_{k-1}^{(n+1)}(\Delta\alpha)}{\tilde{E}_{k-1}^{(n+1)}(\Delta S)} \frac{E_{k-1}^{(n+1)}(g_k)}{\tilde{E}_{k-1}^{(n)}(\Delta g_k)},$$

which is the definition of μ_k^n given by Carstensen.

We have:

PROPERTY 13

$$e_{2k+1}^n = \frac{E_{k+1}^{(n)}(\alpha) - E_k^{(n)}(\alpha)}{E_{k+1}^{(n)}(S) - E_k^{(n)}(S)}.$$

Proof

By the rule of the E-algorithm, we have

$$E_{k+1}^{(n)}(\alpha) - E_k^{(n)}(\alpha) = -\frac{\Delta E_k^{(n)}(\alpha)}{\Delta E_k^{(n)}(g_{k+1})} E_k^{(n)}(g_{k+1})$$

and a similar expression for $E_{k+1}^{(n)}(S) - E_k^{(n)}(S)$. Taking the ratio of these two quantities, the result follows. □

When $\forall n$, $\Delta\alpha_n = 1$ then $\tilde{E}_k^{(n)}(\Delta\alpha) = 1$ and it follows that

$$e_{2k+1}^n = 1 / \tilde{E}_k^{(n)}(\Delta S),$$

which generalizes a well known property of the ε-algorithm of Wynn. Moreover, when $g_i(n) = \Delta^i S_n$, which corresponds to Wynn's algorithm, it is easy to see, from the determinantal expressions given above, that $\mu_k^n = 1$. For another derivation of Wynn's ε-algorithm from the E-algorithm, see [1].

6. Extensions

The preceding approach to the E-algorithm can be easily generalized thus giving rise to new extrapolation procedures. For example, let us replace the operator Δ appearing in the definition of N_k by Δ^2, that is, let us define the new operators M_k by

$$M_0^{(n)}(u) = u_n,$$

$$M_{k+1}^{(n)}(u) = \Delta^2\left(\frac{M_k^{(n)}(u)}{M_k^{(n)}(g_{k+1})}\right),$$

and the new sequence transformation $F_k : (S_n) \mapsto (F_k^{(n)})$ by

$$F_k^{(n)}(S) = \frac{M_k^{(n)}(S)}{M_k^{(n)}(1)}.$$

When $k = 1$ and $g_1(n) = \Delta S_n$, the first column of the Θ-algorithm is recovered, that is

$$F_1^{(n)}(S) = \Theta_2^{(n)}.$$

The other columns of the Θ-algorithm do not seem to be related to this new transformation which should be studied in detail since, as shown by some numerical experiments we conducted, it seems to be quite promising. Obviously, the operator Δ in the definition of N_k could be replaced by Δ^p, where the exponent p can even depend on k.

Another possible extension of the procedure consists in assuming that the sequence $S = (S_n)$ to be transformed satisfies, $\forall n$,

$$S_\infty - S_n = a_1(n)g_1(n) + a_2 g_2(n) + a_3 g_3(n) + \cdots,$$

where $a_2, a_3, \ldots$ are unknown constants and $(a_1(n))$ is an unknown sequence for which a linear difference operator L such that, $\forall n, L(a_1(n)) = 0$ is known. For example, if $a_1(n)$ is a polynomial of degree $k-1$ in n, L is Δ^k. If $a_1(n)$ is a polynomial of degree $k-1$ with respect to an auxiliary sequence x_n, then L is the divided difference operator of order k.

For dealing with this case, we shall define the sequence (L_k) of difference operators by

$$L_0^{(n)}(u) = u_n,$$

$$L_1^{(n)}(u) = L\left(\frac{L_0^{(n)}(u)}{L_0^{(n)}(g_1)}\right),$$

$$L_{k+1}^{(n)}(u) = \Delta\left(\frac{L_k^{(n)}(u)}{L_k^{(n)}(g_{k+1})}\right),$$

and then treat the sequence (S_n) as above, that is, consider the sequence of transformations $D_k : (S_n) \mapsto (D_k^{(n)})$ given by

$$D_k^{(n)}(S) = \frac{L_k^{(n)}(S)}{L_k^{(n)}(1)}.$$

We intend to return to these extensions in a subsequent paper.

Let us mention that the confluent forms of the E-algorithm [2] and of the ε-algorithm [27] can be treated as above. These confluent forms allow one to obtain approximations of

$$S_\infty = \lim_{t \to \infty} f(t)$$

for a function f such that, $\forall t$,

$$S_\infty - f(t) = a_1 g_1(t) + a_2 g_2(t) + \cdots,$$

where the g_i's are given functions taken as $g_i(t) = f^{(i)}(t)$ in the case of the ε-algorithm. The mechanism of this type of continuous prediction algorithm is similar to the mechanism explained in section 2 after replacing the operator Δ by the operator $D = d/dt$. See [7] for an exposition of this type of algorithm.

The approach developed in this paper has also been extended to the vector and matrix cases [8].

Acknowledgements

We would like to thank B. Beckermann and A. Sidi for interesting discussions about the topics developed in this paper. The referees are also thanked for a careful reading of the manuscript.

References

[1] B. Beckermann, A connection between the E-algorithm and the epsilon-algorithm, in: *Numerical and Applied Mathematics*, ed. C. Brezinski (Baltzer, Basel, 1989) pp. 443–446.

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

[3] C. Brezinski, The Mühlbach–Neville–Aitken algorithm and some extensions, BIT 20(1980) 444–451.

[4] C. Brezinski, A survey of iterative extrapolation by the E-algorithm, Det Kong. Norske Vid. Selsk. skr. 2(1989)1–26.

[5] C. Brezinski, Algebraic properties of the E-transformation, in: *Numerical Analysis and Mathematical Modelling*, Banach Center Publications, Vol. 24 (PWN, Warsaw, 1990) pp. 85–90.

[6] C. Brezinski and A.C. Matos, A derivation of extrapolation algorithms based on error estimates, J. Comput. Appl. Math., to appear.

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

[8] C. Brezinski and A. Salam, Matrix and vector sequence transformations revisited, submitted.

[9] C. Brezinski and G. Walz, Sequences of transformations and triangular recursion schemes with applications in numerical analysis, J. Comput. Appl. Math. 34(1991)361–383.

[10] C. Carstensen, On a general epsilon algorithm, in: *Numerical and Applied Mathematics*, ed. C. Brezinski (Baltzer, Basel, 1989) pp. 437–441.

[11] W.F. Ford and A. Sidi, An algorithm for a generalization of the Richardson extrapolation process, SIAM J. Numer. Anal. 24(1987)1212–1232.

[12] T. Håvie, Generalized Neville type extrapolation schemes, BIT 19(1979)204–213.

[13] A.C. Matos and M. Prévost, Acceleration property of the E-algorithm, Numer. Algor. 2(1992)393–408.

[14] G. Meinardus and G.D. Taylor, Lower estimates for the error of best uniform approximation, J. Approx. Theory 16(1976)150–161.

[15] G. Mühlbach, A recurrence formula for generalized divided differences and some applications, J. Approx. Theory 9(1973)165–172.

[16] G. Mühlbach, Newton- und Hermite-Interpolation mit Čebyšev-Systemen, Z. Angew. Math. Mech. 54(1974)541–550.

[17] G. Mühlbach, Neville–Aitken algorithms for interpolation by functions of Čebyšev-systems in the sense of Newton and in a generalized sense of Hermite, in: *Theory of Approximation, with Applications*, ed. A.G. Law and B.N. Sahney (Academic Press, New York, 1967) pp. 200–212.

[18] G. Mühlbach, The general Neville–Aitken algorithm and some applications, Numer. Math. 31(1978)97–110.

[19] G. Mühlbach, The general recurrence relation for divided differences and the general Newton-interpolation-algorithm with applications to trigonometric interpolation, Numer. Math. 32(1979) 393–408.

[20] M. Prévost, Acceleration property for the E-algorithm and an application to the summation of series, Adv. Comput. Math. 2(1994)319–341.

[21] C. Schneider, Vereinfachte Rekursionen zur Richardson-Extrapolation in Spezialfällen, Numer. Math. 24(1975)177–184.

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

[23] A. Sidi, An algorithm for a special case of a generalization of the Richardson extrapolation process, Numer. Math. 38(1982)299–307.

[24] A. Sidi, On a generalization of the Richardson extrapolation process, Numer. Math. 57(1990) 365–377.

[25] E.J. Weniger, Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series, Comput. Phys. Rep. 10(1989)189–371.

[26] P. Wynn, On a device for computing the $e_m(S_n)$ transformation, MTAC 10(1956)91–96.

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