ELSEVIER

Applied Numerical Mathematics 16 (1994) 239-244

APPLIED NUMERICAL MATHEMATICS

On the kernel of sequence transformations

C. Brezinski [a,*], M. Redivo-Zaglia [b,1]

[a] *Laboratoire d'Analyse Numérique et d'Optimisation, UFR IEEA – M3, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq cedex, France*

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

This paper is dedicated to Professor Robert Vichnevetsky to honor him on the occasion of his 65th birthday

Abstract

In this paper, the kernels (that is the sets of sequences transformed into a constant sequence) of the sequence transformations where each step is obtained from the previous one are studied in detail. It is first proved that the kernel of each step contains the kernel of the previous step. Then, a technique for constructing the difference equation satisfied by the sequences of the kernel is given. It is illustrated by several examples.

Keywords: Extrapolation methods; Convergence acceleration

1. Introduction

The construction of extrapolation algorithms by using annihilation difference operators was initiated by Weniger [8]. It was extended and developed in [3]. This approach is a very general one since it covers all the algorithms currently known. Our aim, in this paper, is to focus on the kernel of the sequence transformations which are obtained by using successively several difference operators.

Sequences will be denoted by letters and their terms by the same letter with a subscript. If $u = (u_n)$ and $v = (v_n)$, we shall make use of the notation $u/v = (u_n/v_n)$ and of a similar notation for the product.

Let us first recall some definitions.

* Corresponding author. E-mail: brezinsk@omega.univ-lille1.fr.

[1] E-mail: elen@elett1.dei.unipd.it.

SSDI 0168-9274(94)00054-9

Definition 1 [9, p. 196]. Let $\mathscr{S}$ be the set of complex sequences. A *difference operator* L is a linear mapping of $\mathscr{S}$ into itself

$$L : u = (u_n) \in \mathscr{S} \mapsto L(u) = ((L(u))_n) \in \mathscr{S}.$$

The nth term $(L(u))_n$ of the sequence $L(u)$ will be denoted by $L_n(u)$ or, sometimes, by $L(u_n)$.

The most general difference operator is given by

$$L_n(u) = \sum_{i=-p_n}^{q_n} G_i(n) u_{n+i},$$

where p_n and q_n are nonnegative integers which may depend on n, $u_i = 0$ for $i < 0$, and the G_i's are given functions of n which can also depend on auxiliary fixed sequences. It should be clearly understood that, if these auxiliary sequences also depend on some terms of the sequence (u_n) itself, then these terms are *fixed* in the G_i's and thus the operator L is still linear.

Definition 2 [8, p. 212]. L is called an *annihilation difference operator for the sequence* $a = (a_n)$ if there exists N such that for each $n \geqslant N$, $L_n(a) = 0$.

Definition 3 [8, p. 212]. The sequence (D_n) is called a *remainder* (*or error*) *estimate of the sequence* (S_n) if for all n, $S_\infty - S_n = a_n D_n$ where (a_n) is an unknown sequence and S_∞ a (usually unknown) number. If (S_n) converges to S_∞, then S_∞ is called its *limit* and, otherwise, its *antilimit*.

Let us assume that the sequence $S = (S_n)$ satisfies

$$\forall n, \; S_\infty - S_n = a_n D_n,$$

where $a = (a_n)$ is an unknown sequence and $D = (D_n)$ a known one. Let L be an annihilation difference operator for a.

We wish to construct a sequence transformation $\mathscr{T} : (S_n) \mapsto (T_n)$ such that $\exists N$, $\forall n \geqslant N$, $T_n = S_\infty$. Thus, we have

$$\frac{S_\infty}{D_n} - \frac{S_n}{D_n} = a_n.$$

Applying L to both sides of this relation and using its linearity property, we obtain, $\forall n \geqslant N$,

$$S_\infty L_n(1/D) - L_n(S/D) = L_n(a) = 0,$$

and it follows that

$$S_\infty = \frac{L_n(S/D)}{L_n(1/D)}.$$

Thus, if the transformation $\mathcal{T}:(S_n) \mapsto (T_n)$ is defined by

$$T_n = \frac{L_n(S/D)}{L_n(1/D)},$$

then, by construction, $\forall n \geqslant N$, $T_n = S_\infty$ if and only if $\forall n \geqslant N$, $S_\infty - S_n = a_n D_n$.

We recall the following definition:

Definition 4 [4, p. 4]. Let $\mathcal{T}:(S_n) \mapsto (T_n)$ be a sequence transformation. The kernel $\mathcal{K}_\mathcal{T}$ of $\mathcal{T}$ is the set of sequences such that $\exists S_\infty$, $\exists N$, $\forall n \geqslant N$, $T_n = S_\infty$.

In the sequel, we shall assume, without any loss of generality, that $N = 0$.

2. Composition of transformations

In this section, for convenience, we shall denote by $L(u_n)$ the nth term of the sequence $L(u)$. We shall use several difference operators denoted by $L^{(i)}$ and we shall compose the transformations defined by the difference operators $L^{(i)}$, that is, in other terms, we have the following iterative use of the procedure.

- We set

$$T_0^{(n)} = S_n \text{ and } D_0^{(n)} = D_n.$$

- Then, for $k = 0, 1, \ldots,$ we set

$$T_{k+1}^{(n)} = \frac{L^{(k+1)}\left(T_k^{(n)}/D_k^{(n)}\right)}{L^{(k+1)}\left(1/D_k^{(n)}\right)},$$

where the $(D_k^{(n)})$ are given sequences.

Let $\mathcal{T}_k$ be the transformation $(S_n) \mapsto (T_k^{(n)})$ for a fixed value of k. It corresponds to a composition of the transformations defined by the successive operators $L^{(1)}, L^{(2)}, \ldots, L^{(k)}$.

We shall study the kernels $\mathcal{N}_k$ of these successive transformations $\mathcal{T}_k$. We first have the following result.

Theorem 5. *The kernel of $\mathcal{T}_{k+1}$ contains the kernel of $\mathcal{T}_k$.*

Proof. Let us assume that $(S_n) \in \mathcal{N}_k$. Then, $\forall n$, $T_k^{(n)} = S_\infty$ and thus

$$T_{k+1}^{(n)} = \frac{L^{(k+1)}\left(T_k^{(n)}/D_k^{(n)}\right)}{L^{(k+1)}\left(1/D_k^{(n)}\right)} = \frac{S_\infty L^{(k+1)}\left(1/D_k^{(n)}\right)}{L^{(k+1)}\left(1/D_k^{(n)}\right)} = S_\infty. \qquad \square$$

We set $a_0^{(n)} = a_n$ and, for $k = 1, 2, \ldots,$

$$b_k^{(n)} = \frac{1}{D_k^{(n)} L^{(k)}\left(1/D_{k-1}^{(n)}\right)}, \qquad a_k^{(n)} = b_k^{(n)} L^{(k)}\left(a_{k-1}^{(n)}\right).$$

Let us now characterize the kernels $\mathcal{N}_k$.

Theorem 6. *With the previous notation*

$$\mathcal{N}_k = \{(S_n),\ \forall n,\ S_\infty - S_n = a_n D_n \quad \text{with } L^{(k)}\big(b_{k-1}^{(n)} L^{(k-1)}\big(\cdots L^{(2)}\big(b_1^{(n)} L^{(1)}(a_n)\big)\cdots\big)\big) = 0\}.$$

Proof. By construction, we have

$$S_\infty - T_k^{(n)} = a_k^{(n)} D_k^{(n)} = \frac{L^{(k)}\big(a_{k-1}^{(n)}\big)}{L^{(k)}\big(1/D_{k-1}^{(n)}\big)}$$

and

$$\mathcal{N}_k = \{(S_n),\ \forall n,\ S_\infty - T_{k-1}^{(n)} = a_{k-1}^{(n)} D_{k-1}^{(n)} \text{ with } L^{(k)}\big(a_{k-1}^{(n)}\big) = 0\}.$$

It is easy to see, by applying the definition of $a_{k-1}^{(n)}$, that $\mathcal{N}_k$ can also be written, for $k \geqslant 2$, as

$$\mathcal{N}_k = \{(S_n),\ \forall n,\ S_\infty - T_{k-2}^{(n)} = a_{k-2}^{(n)} D_{k-2}^{(n)} \text{ with } L^{(k)}\big(b_{k-1}^{(n)} L^{(k-1)}\big(a_{k-2}^{(n)}\big)\big) = 0\}.$$

By using these relations in a recursive way (that is by replacing, in the last relation, $a_{k-2}^{(n)}$ by its expression in terms of the previous quantities and iterating this procedure until $k = 0$), the kernel $\mathcal{N}_k$ of the transformation $\mathcal{T}_k$ can be expressed, as in the theorem, by a difference equation with respect to the sequence (a_n). □

For the choice $D_k^{(n)} = 1/L^{(k)}(1/D_{k-1}^{(n)})$ proposed in [3], we have $b_k^{(n)} = 1$ and thus this difference equation becomes $L^{(k)}(\cdots L^{(1)}(a_n)\cdots) = 0$.

Solving the difference equation given in this theorem furnishes the sequences belonging to $\mathcal{N}_k$ under the form $S_\infty - S_n = a_n D_n$ with a closed expression, depending on the $(D_i^{(n)})$, for a_n. If the $D_i^{(n)}$ depend on the sequence (S_n) itself, then $S_\infty - S_n = a_n D_n$ is a difference equation which has to be solved to get a closed expression for the sequences (S_n) belonging to $\mathcal{N}_k$. Usually, it is a nonlinear difference equation which explains why it has been solved only in a few particular cases.

We shall now illustrate with several examples this technique for obtaining the kernels.

3. Applications

The Θ-algorithm was introduced in [1]. Its first step consists, in fact, of constructing the sequence

$$\Theta_2^{(n)} = \frac{\Delta^2(S_n/\Delta S_n)}{\Delta^2(1/\Delta S_n)}.$$

From Theorem 6, its kernel is the set of sequences such that $\forall n$, $S_\infty - S_n = a_n \Delta S_n$ with a_n satisfying $\Delta^2 a_n = 0$. This is a very simple case and, solving this difference equation, leads to $\Delta a_n = c$ where c is a constant. In other words, the kernel of this transformation is the set of sequences such that

$$\forall n,\ \Delta((S_\infty - S_n)/\Delta S_n) = c,$$

which is the relation obtained by Cordellier [6] who got the solution of this nonlinear difference equation by solving successively two linear difference equations. It should also be noticed that the solution of the difference equation $\Delta a_n = c$ is $a_n = cn + d$ where d is a constant. Thus the kernel of the first column of the Θ-algorithm is the set of sequences such that

$$\forall n,\ S_\infty - S_n = (cn + d)\Delta S_n.$$

In [5], a generalization of the Θ-algorithm was proposed. Its first step is similar to the first step of the Θ-algorithm but with an arbitrary sequence $(g_1(n))$ instead of (ΔS_n). So, its kernel is the set of sequences such that

$$\forall n,\ S_\infty - S_n = (cn + d)g_1(n).$$

Let us now consider the general case described in the previous section when $\forall i$, $L^{(i)} = \Delta$. It corresponds to a repeated use of the Θ-procedure [2] and we shall treat the case $k = 2$. By Theorem 6, the kernel of the transformation $\mathscr{T}_2$ is the set of sequences of the form $S_\infty - S_n = a_n D_n$ with (a_n) satisfying the difference equation

$$\Delta\left(\frac{1}{D_1^{(n)}}\frac{\Delta a_n}{\Delta\left(1/D_0^{(n)}\right)}\right) = 0$$

with $D_0^{(n)} = D_n$. Solving this difference equation gives

$$\Delta a_n = cD_1^{(n)}\Delta\left(1/D_0^{(n)}\right),$$

where c is an arbitrary constant.

Let us show how to solve this equation.

For $(S_n) \in \mathscr{N}_2$, we also have

$$-\Delta S_n = \Delta(a_n D_n).$$

Now, using the relation

$$\Delta(a_n D_n) = D_{n+1}\Delta a_n + a_n \Delta D_n,$$

we obtain

$$a_n = c\frac{D_1^{(n)}}{D_0^{(n)}} - \frac{\Delta S_n}{\Delta D_0^{(n)}}.$$

Thus we have the following result

Theorem 7. *The kernel of $\mathscr{T}_2$ is the set of sequences satisfying, $\forall n$,*

$$S_\infty - S_n = cD_1^{(n)} - \frac{\Delta S_n}{\Delta D_0^{(n)}} D_0^{(n)}$$

or

$$(S_\infty - S_n)\frac{D_0^{(n+1)}}{\Delta D_0^{(n)}} - (S_\infty - S_{n+1})\frac{D_0^{(n)}}{\Delta D_0^{(n)}} = cD_1^{(n)}.$$

There are many algorithms that fit into the case $L^{(i)} = \Delta$ and, in particular, Overholt's process [7], which corresponds to $D_0^{(n)} = \Delta S_n$ and $D_1^{(n)} = (\Delta S_{n+1})^2$. Thus its kernel is given in an implicit form as the solution of a nonlinear difference equation whose solution is (for the moment) unknown.

It must be noticed that, even if $L^{(1)} = L^{(2)}$, commutativity does not hold. If, for example, $L^{(1)} = L^{(2)} = \Delta$, then we obtain the following kernels ((x_n) is a given auxiliary sequence):

- With $D_0^{(n)} = (-1)^n$ and $D_1^{(n)} = x_n$

$$S_n - S_\infty = (-1)^n a + u_n,$$

where (u_n) satisfies $u_{n+1} = -u_n - 2cx_n$.

- With $D_0^{(n)} = x_n$ and $D_1^{(n)} = (-1)^n$

$$S_n - S_\infty = ax_n + v_n,$$

where (v_n) satisfies $v_{n+1} = v_n x_{n+1}/x_n + c(-1)^n \Delta x_n / x_n$.

It is easy to see that, if $x_n = (-1)^n$, both kernels are the same.

Acknowledgement

We would like to thank the referees for improving our style.

References

[1] C. Brezinski, Accélération de suites à convergence logarithmique, *C.R. Acad. Sci. Paris* 273 A (1971) 727–730.
[2] C. Brezinski, Some new convergence acceleration methods, *Math. Comput.* 39 (1982) 133–145.
[3] C. Brezinski and A.C. Matos, A derivation of extrapolation algorithms based on error estimates. *J. Comput. Appl. Math.* (to appear).
[4] C. Brezinski and M. Redivo-Zaglia, *Extrapolation Methods. Theory and Practice* (North-Holland, Amsterdam, 1991).
[5] C. Brezinski and M. Redivo-Zaglia, A general extrapolation procedure revisited, *Adv. Comput. Math.* 2 (1994) 461–477.
[6] F. Cordellier, Caractérisation des suites que la première étape du θ-algorithme transforme en suites constantes, *C.R. Acad. Sci. Paris* 284 A (1977) 389–392.
[7] K.J. Overholt, Extended Aitken acceleration, *BIT* 5 (1965) 122–132.
[8] E.J. Weniger, Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series, *Computer Physics Reports* 10 (1989) 189–371.
[9] J. Wimp, *Computation with Recurrence Relations* (Pitman, Boston, MA, 1984).