SIAM J. Sci. Comput.
Vol. 36, No. 5, pp. A2227–A2247

THE SIMPLIFIED TOPOLOGICAL ε-ALGORITHMS FOR ACCELERATING SEQUENCES IN A VECTOR SPACE*

CLAUDE BREZINSKI† AND MICHELA REDIVO-ZAGLIA‡

Abstract. When a sequence of numbers is slowly converging, it can be transformed into a new sequence which, under some assumptions, converges faster to the same limit. One of the best-known sequence transformations is the Shanks transformation, which can be recursively implemented by the ε-algorithm of Wynn. This transformation and this algorithm have been extended (in two different ways) to a sequence of elements of a topological vector space E. In this paper, we present new algorithms for implementing these topological Shanks transformations. They no longer require the manipulation of elements of the algebraic dual space E^* of E, nor do they use the duality product inside the rules of the algorithms; they need the storage of fewer elements of E, and the stability is improved. They also allow us to prove convergence and acceleration results for some types of sequences. Various applications involving sequences of vectors or matrices show the interest of the new algorithms.

Key words. convergence acceleration, extrapolation, matrix equations, matrix functions, iterative methods

AMS subject classifications. 65B05, 65F10, 65F30, 65F60, 65J10, 65J15, 65Q20

DOI. 10.1137/140957044

1. Presentation.

Let $(\mathbf{S}_n)$ be a sequence of elements of a vector space E on a field $\mathbb{K}$ ($\mathbb{R}$ or $\mathbb{C}$). If the sequence $(\mathbf{S}_n)$ is slowly converging to its limit $\mathbf{S}$ when n tends to infinity, it can be transformed, by a *sequence transformation*, into a new sequence converging, under some assumptions, faster to the same limit. The construction of most sequence transformations starts from the notion of *kernel*, which is the set of sequences which are transformed into a constant sequence with all its elements equal to $\mathbf{S}$ (see [9]).

In section 2, the scalar Shanks transformation for transforming a sequence of numbers [31] and its implementation by the scalar ε-algorithm [36] are remembered. This transformation and this algorithm were extended by Brezinski [3] to sequences of elements of a topological vector space E. The two versions of this *topological Shanks transformation* are given in section 3. Their original implementation by two types of the *topological ε-algorithm* is discussed in section 4. Both algorithms need to perform operations involving elements of the algebraic dual space E^* of E.

The topological Shanks transformations and the topological ε-algorithms received so much attention in the literature that not all of the contributions could be quoted here. They have many applications in the solution of systems of linear and nonlinear equations, in the computation of eigenelements, in Padé-type approximation, in matrix functions and matrix equations, in the Lanczos method, etc. We refer the reader

*Submitted to the journal's Methods and Algorithms for Scientific Computing section February 13, 2014; accepted for publication (in revised form) July 1, 2014; published electronically September 23, 2014.
http://www.siam.org/journals/sisc/36-5/95704.html

†Laboratoire Paul Painlevé, UMR CNRS 8524, UFR de Mathématiques, Université des Sciences et Technologies de Lille, 59655 Villeneuve d'Ascq cedex, France (Claude.Brezinski@univ-lille1.fr). This author's work was partially supported by the Labex CEMPI (ANR–11–LABX–0007–01).

‡Dipartimento di Matematica, Università degli Studi di Padova, 35121 Padova, Italy (Michela.RedivoZaglia@unipd.it). This author's work was partially supported by the University of Padova, Project 2012 CPDA124755, and by GNCS-INdAM.

to [6, 19, 30, 9, 20, 21, 11, 18, 32, 33, 35] and the references therein, in particular, for the older ones.

Two new algorithms for implementing these topological Shanks transformations, the first and the second *simplified topological ε-algorithms*, are introduced in section 5. They have several important advantages: elements of the dual vector space E^* no longer have to be used in their recursive rules and are replaced by quantities computed by the scalar ε-algorithm, fewer elements have to be stored than with the topological ε-algorithms of [3], a very important issue in applications, and the numerical stability of the algorithms is improved. Moreover, these new algorithms allow us to prove convergence and acceleration properties that could hardly have been obtained from the original topological ε-algorithms (section 6). The implementation of these algorithms is discussed in section 7. Some applications involving sequences of vectors and matrices are presented in section 8.

In this paper, characters in bold correspond to elements of a vector space E or its algebraic dual E^*, while regular ones designate real or complex numbers.

2. Shanks transformation and the ε-algorithm.

For sequences (S_n) of real or complex numbers, an important sequence transformation is the Shanks transformation [31], whose kernel consists of sequences satisfying the homogeneous linear difference equation of order k,

$$a_0(S_n - S) + \cdots + a_k(S_{n+k} - S) = 0, \quad n = 0, 1, \ldots, \tag{2.1}$$

where the a_i's are arbitrary constants independent of n such that $a_0 a_k \neq 0$, and $a_0 + \cdots + a_k \neq 0$, and where the unknown S is the limit of (S_n) if it converges, or is called its antilimit otherwise. The exact expression of such sequences is given in [8]. Assuming that (2.1) holds for all n, the problem is to compute S. We have, for all n,

$$a_0 \Delta S_n + \cdots + a_k \Delta S_{n+k} = 0, \tag{2.2}$$

where the forward difference operator Δ is defined by $\Delta S_n = S_{n+1} - S_n$, $n = 0, 1, \ldots$. Writing (2.2) for the indexes $n, \ldots, n+k-1$ leads to k homogeneous linear equations in the $k+1$ unknowns $a_0, \ldots, a_k$. Adding the condition $a_0 + \cdots + a_k = 1$ (which does not restrict the generality), and solving the system obtained, gives the unknowns, and then, from (2.1), we obtain, for all n,

$$S = a_0 S_n + \cdots + a_k S_{n+k}.$$

Now, if (S_n) does not satisfy (2.1), we can still write down the system giving the unknown coefficients a_i and compute the linear combination $a_0 S_n + \cdots + a_k S_{n+k}$. However, these coefficients and the linear combination will now depend on n and k, and they will be denoted, respectively, by $a_i^{(n,k)}$ and $e_k(S_n)$, and we have

$$e_k(S_n) = a_0^{(n,k)} S_n + \cdots + a_k^{(n,k)} S_{n+k}, \quad k, n = 0, 1, \ldots,$$

where the $a_i^{(n,k)}$'s are solution of the same system as above, that is,

$$\left\{ \begin{array}{ccccccc} a_0^{(n,k)} & + & \cdots & + & a_k^{(n,k)} & = & 1, \\ a_0^{(n,k)} \Delta S_n & + & \cdots & + & a_k^{(n,k)} \Delta S_{n+k} & = & 0, \\ \vdots & & & & \vdots & & \\ a_0^{(n,k)} \Delta S_{n+k-1} & + & \cdots & + & a_k^{(n,k)} \Delta S_{n+2k-1} & = & 0. \end{array} \right. \tag{2.3}$$

Thus, the sequence (S_n) has been transformed into the set of sequences $\{(e_k(S_n))\}$. The transformation $(S_n) \longmapsto \{(e_k(S_n))\}$ is the *Shanks transformation* [31]. It is a generalization of the well-known Aitken's Δ^2 process which is recovered for $k = 1$.

Cramer's rule allows us to write $e_k(S_n)$ as the following ratio of determinants:

$$e_k(S_n) = \frac{\begin{vmatrix} S_n & \cdots & S_{n+k} \\ \Delta S_n & \cdots & \Delta S_{n+k} \\ \vdots & & \vdots \\ \Delta S_{n+k-1} & \cdots & \Delta S_{n+2k-1} \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ \Delta S_n & \cdots & \Delta S_{n+k} \\ \vdots & & \vdots \\ \Delta S_{n+k-1} & \cdots & \Delta S_{n+2k-1} \end{vmatrix}}, \quad k, n = 0, 1, \ldots.$$

By construction, we have the following result.

THEOREM 2.1 (see [9]). *For all n, $e_k(S_n) = S$ if and only if $\exists a_0, \ldots, a_k$, with $a_0 a_k \neq 0$ and $a_0 + \cdots + a_k \neq 0$, such that, for all n,*

$$a_0(S_n - S) + \cdots + a_k(S_{n+k} - S) = 0,$$

that is, in other words, if and only if (S_n) belongs to the kernel of the Shanks transformation $(S_n) \longmapsto (e_k(S_n))_n$ for k fixed.

The Shanks transformation can be recursively implemented by the scalar ε-algorithm of Wynn [36], whose rules are

$$\text{(2.4)} \qquad \begin{cases} \varepsilon_{-1}^{(n)} = 0, & n = 0, 1, \ldots, \\ \varepsilon_{0}^{(n)} = S_n, & n = 0, 1, \ldots, \\ \varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + (\varepsilon_{k}^{(n+1)} - \varepsilon_{k}^{(n)})^{-1}, & k, n = 0, 1, \ldots, \end{cases}$$

and the following property holds.

PROPERTY 2.2. *For all k and n, $\varepsilon_{2k}^{(n)} = e_k(S_n)$ and $\varepsilon_{2k+1}^{(n)} = 1/e_k(\Delta S_n)$.*

These elements are usually displayed in a two-dimensional array, the ε-array (see Figure 1 for a triangular part of it), where the quantities with an odd lower index are only intermediate results. The rule (2.4) relates numbers located at the four vertices of a rhombus in the ε-array (see Figure 1).

In passing, let us give two relations which do not seem to have been noticed earlier. Their proofs are straightforward by induction.

PROPERTY 2.3. *For all k and n,*

$$\varepsilon_{2k}^{(n)} = S_{n+k} + \sum_{i=1}^{k} 1/\Delta\varepsilon_{2i-1}^{(n+k-i)}, \qquad \varepsilon_{2k+1}^{(n)} = \sum_{i=0}^{k} 1/\Delta\varepsilon_{2i}^{(n+k-i)},$$

where the operator Δ acts on the upper indexes.

3. The topological Shanks transformations.

Let us now consider the case of a sequence of elements $\mathbf{S}_n$ of a vector space E. In 1962, Wynn generalized the scalar ε-algorithm to sequences of vectors $(\mathbf{S}_n)$ by defining, in the rule of the scalar ε-algorithm, the inverse of a vector $\mathbf{u} \in E = \mathbb{R}^m$ by $\mathbf{u}^{-1} = \mathbf{u}/(\mathbf{u}, \mathbf{u})$, where $(\cdot, \cdot)$ is the usual inner product [37]. He thus obtained the *vector ε-algorithm* (VEA), which

$$
\begin{array}{llllll}
\varepsilon_{-1}^{(0)} = 0 & & & & & \\
 & \varepsilon_0^{(0)} = S_0 & & & & \\
\varepsilon_{-1}^{(1)} = 0 & & \varepsilon_1^{(0)} & & & \\
 & \varepsilon_0^{(1)} = S_1 & & \varepsilon_2^{(0)} & & \\
\varepsilon_{-1}^{(2)} = 0 & & \varepsilon_1^{(1)} & & \varepsilon_3^{(0)} & \\
 & \varepsilon_0^{(2)} = S_2 & & \varepsilon_2^{(1)} & & \varepsilon_4^{(0)} \\
\varepsilon_{-1}^{(3)} = 0 & & \varepsilon_1^{(2)} & & \varepsilon_3^{(1)} & \\
 & \varepsilon_0^{(3)} = S_3 & & \varepsilon_2^{(2)} & & \\
\varepsilon_{-1}^{(4)} = 0 & & \varepsilon_1^{(3)} & & & \\
 & \varepsilon_0^{(4)} = S_4 & & & & \\
\varepsilon_{-1}^{(5)} = 0 & & & & &
\end{array}
$$

FIG. 1. *A triangular part of the ε-array.*

had no underlying algebraic foundation. However, it was later proved by McLeod [24] that the kernel of this vector ε-algorithm is the set of vector sequences satisfying a relation of the same form as (2.1) with $a_i \in \mathbb{R}$. The proof was quite technical, and it involved Clifford algebra. Then, it was shown that the vectors $\boldsymbol{\varepsilon}_{2k}^{(n)} \in \mathbb{R}^m$ it computes are given by ratios of determinants of dimension $2k+1$ instead of $k+1$ as in the scalar case [17], or by designants of dimension $k+1$, objects that generalize determinants in a noncommutative algebra [29].

To remedy the lack, for the vector ε-algorithm, of an algebraic theory similar to that existing for the scalar one, Brezinski, following the approach of Shanks, obtained two versions of the so-called *topological Shanks transformation* [3]. More generally, they can be applied to sequences of elements of a topological vector space E (thus the name) on $\mathbb{K}$ ($\mathbb{R}$ or $\mathbb{C}$) since, for being able to deal with convergence results, E has to possess a topology. Starting from a relation of the same form as (2.1), that is,

$$a_0(\mathbf{S}_n - \mathbf{S}) + \cdots + a_k(\mathbf{S}_{n+k} - \mathbf{S}) = \mathbf{0} \in E, \quad n = 0, 1, \ldots, \tag{3.1}$$

where $\mathbf{S}_n, \mathbf{S} \in E$ and $a_i \in \mathbb{K}$ with $a_0 a_k \neq 0$ and $a_0 + \cdots + a_k \neq 0$, we again write

$$a_0 \Delta\mathbf{S}_n + \cdots + a_k \Delta\mathbf{S}_{n+k} = \mathbf{0}$$

for the indexes $n, \ldots, n+k-1$. However, these relations do not allow us to compute the numbers a_i since the $\mathbf{S}_n$ are elements of E (in many practical applications they are vectors of $\mathbb{R}^m$ or matrices of dimension $m \times s$). For that purpose, let $\mathbf{y} \in E^*$ (the algebraic dual space of E, that is, the vector space of linear functionals on E), and take the duality product of these relations with it. We thus obtain

$$a_0\langle \mathbf{y}, \Delta\mathbf{S}_i\rangle + \cdots + a_k\langle \mathbf{y}, \Delta\mathbf{S}_{i+k}\rangle = \mathbf{0}, \qquad i = n, \ldots, n+k-1. \tag{3.2}$$

As in the scalar case, if $(\mathbf{S}_n)$ does not satisfy (3.1), the preceding relations, together with the additional condition that the coefficients a_i sum up to 1, can still be written. We thus obtain a system of $k+1$ equations in $k+1$ unknowns, denoted by $a_i^{(n,k)}$, similar to (2.3),

$$\left\{\begin{array}{ccccccc}
a_0^{(n,k)} & + & \cdots & + & a_k^{(n,k)} & = & 1, \\
a_0^{(n,k)}\langle \mathbf{y}, \Delta\mathbf{S}_n\rangle & + & \cdots & + & a_k^{(n,k)}\langle \mathbf{y}, \Delta\mathbf{S}_{n+k}\rangle & = & 0, \\
\vdots & & & & \vdots & & \\
a_0^{(n,k)}\langle \mathbf{y}, \Delta\mathbf{S}_{n+k-1}\rangle & + & \cdots & + & a_k^{(n,k)}\langle \mathbf{y}, \Delta\mathbf{S}_{n+2k-1}\rangle & = & 0,
\end{array}\right. \tag{3.3}$$

and we define the first topological Shanks transformation by

$$\widehat{\mathbf{e}}_k(\mathbf{S}_n) = a_0^{(n,k)}\mathbf{S}_n + \cdots + a_k^{(n,k)}\mathbf{S}_{n+k}, \quad n, k = 0, 1, \ldots .$$

Thus, as in the scalar case, we have

$$\widehat{\mathbf{e}}_k(\mathbf{S}_n) = \frac{\begin{vmatrix} \mathbf{S}_n & \cdots & \mathbf{S}_{n+k} \\ \langle \mathbf{y}, \Delta\mathbf{S}_n \rangle & \cdots & \langle \mathbf{y}, \Delta\mathbf{S}_{n+k} \rangle \\ \vdots & & \vdots \\ \langle \mathbf{y}, \Delta\mathbf{S}_{n+k-1} \rangle & \cdots & \langle \mathbf{y}, \Delta\mathbf{S}_{n+2k-1} \rangle \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ \langle \mathbf{y}, \Delta\mathbf{S}_n \rangle & \cdots & \langle \mathbf{y}, \Delta\mathbf{S}_{n+k} \rangle \\ \vdots & & \vdots \\ \langle \mathbf{y}, \Delta\mathbf{S}_{n+k-1} \rangle & \cdots & \langle \mathbf{y}, \Delta\mathbf{S}_{n+2k-1} \rangle \end{vmatrix}}, \quad k, n = 0, 1, \ldots,$$

where the determinant in the numerator denotes the element of E obtained by developing it with respect to its first row by the classical rule for expanding a determinant.

Similarly, the second topological Shanks transformation is defined by

$$\widetilde{\mathbf{e}}_k(\mathbf{S}_n) = a_0^{(n,k)}\mathbf{S}_{n+k} + \cdots + a_k^{(n,k)}\mathbf{S}_{n+2k}, \quad n, k = 0, 1, \ldots,$$

where the $a_i^{(n,k)}$'s are again solution of the system (3.3) and, thus, are identical to the coefficients in the first topological Shanks transformation. These $\widetilde{\mathbf{e}}_k(\mathbf{S}_n)$ are given by the same ratio of determinants as above after replacing the first row in the numerator by $\mathbf{S}_{n+k}, \ldots, \mathbf{S}_{n+2k}$.

Obviously, the linear functional $\mathbf{y}$ must be chosen so that the system (3.3) is nonsingular for the values of k and n considered. This condition will always be assumed in what follows, thus implying that the results hold for all such $\mathbf{y}$. The choice of $\mathbf{y}$ in the vector and matrix cases will be discussed in section 7.

The following fundamental property shows the connection between the scalar and the topological Shanks transformations.

PROPERTY 3.1. *Setting $S_n = \langle \mathbf{y}, \mathbf{S}_n \rangle$, the systems* (2.3) *and* (3.3) *giving the coefficients $a_i^{(n,k)}$ of the scalar, the first, and the second topological Shanks transformations are the same.*

We will see below that this property is a quite fundamental one, having important consequences for the convergence and the acceleration properties of the topological Shanks transformations.

For the first and the second topological Shanks transformations, the following theorem holds.

THEOREM 3.2 (see [3]). *For all n, $\widehat{\mathbf{e}}_k(\mathbf{S}_n) = \mathbf{S}$ and $\widetilde{\mathbf{e}}_{\mathbf{k}}(\mathbf{S}_n) = \mathbf{S}$ if $\exists a_0, \ldots, a_k$, with $a_0 a_k \neq 0$ and $a_0 + \cdots + a_k \neq 0$, such that, for all n,*

$$a_0(\mathbf{S}_n - \mathbf{S}) + \cdots + a_k(\mathbf{S}_{n+k} - \mathbf{S}) = \mathbf{0}.$$

Remark 3.3. Contrarily to Theorem 2.1, the condition of this theorem is only sufficient. If the condition is satisfied, we also have, for all n,

$$a_0\langle \mathbf{y}, \mathbf{S}_n - \mathbf{S} \rangle + \cdots + a_k\langle \mathbf{y}, \mathbf{S}_{n+k} - \mathbf{S} \rangle = 0.$$

Let us mention that other transformations based on a kernel of the form (3.1) have been studied in the case where $E = \mathbb{R}^m$ or $\mathbb{R}^{m\times s}$ (see, for example, [9, 20]). The idea is always to obtain a system of linear algebraic equations allowing one to compute the coefficients a_i appearing in (3.1). In these methods, the relations (3.2) are replaced by

$$a_0\langle \mathbf{y}_{i,0}, \Delta\mathbf{S}_n\rangle + \cdots + a_k\langle \mathbf{y}_{i,k}, \Delta\mathbf{S}_{n+k}\rangle = 0, \qquad i = 0,\ldots,k-1,$$

with $\mathbf{y}_{i,j} = \Delta\mathbf{S}_{n+i}$ for the *Reduced Rank Extrapolation* (RRE), $\mathbf{y}_{i,j} = \Delta\mathbf{S}_{n+i+1} - \Delta\mathbf{S}_{n+i}$ for the *Minimal Polynomial Extrapolation* (MPE), and $\mathbf{y}_{i,j} = \mathbf{y}_{i+1}$ (arbitrary linearly independent linear functionals) for the *Modified Minimal Polynomial Extrapolation* (MMPE). These methods were recently studied in [21], including the first topological Shanks transformation.

4. The topological ε-algorithms.

Let us now discuss how to compute recursively the elements $\widehat{\mathbf{e}}_k(\mathbf{S}_n) \in E$ and $\widetilde{\mathbf{e}}_k(\mathbf{S}_n) \in E$.

The elements $\widehat{\mathbf{e}}_k(\mathbf{S}_n) \in E$ can be recursively computed by the *first topological ε-algorithm* (TEA1) [3], whose rules are

$$(4.1)\quad \begin{cases} \widehat{\varepsilon}_{-1}^{(n)} &= \mathbf{0} \in E^*, \qquad n = 0,1,\ldots, \\ \widehat{\varepsilon}_{0}^{(n)} &= \mathbf{S}_n \in E, \qquad n = 0,1,\ldots, \\ \widehat{\varepsilon}_{2k+1}^{(n)} &= \widehat{\varepsilon}_{2k-1}^{(n+1)} + \dfrac{\mathbf{y}}{\langle \mathbf{y}, \widehat{\varepsilon}_{2k}^{(n+1)} - \widehat{\varepsilon}_{2k}^{(n)}\rangle} \in E^*, \quad k,n = 0,1,\ldots, \\ \widehat{\varepsilon}_{2k+2}^{(n)} &= \widehat{\varepsilon}_{2k}^{(n+1)} + \dfrac{\widehat{\varepsilon}_{2k}^{(n+1)} - \widehat{\varepsilon}_{2k}^{(n)}}{\langle \widehat{\varepsilon}_{2k+1}^{(n+1)} - \widehat{\varepsilon}_{2k+1}^{(n)}, \widehat{\varepsilon}_{2k}^{(n+1)} - \widehat{\varepsilon}_{2k}^{(n)}\rangle} \in E, \quad k,n = 0,1,\ldots. \end{cases}$$

Let us recall that the idea which led to the discovery of the two topological ε-algorithms for implementing the two versions of the topological Shanks sequence transformation was based on the definition of the inverse of a couple $(\mathbf{u},\mathbf{y}) \in E \times E^*$ defined as $\mathbf{u}^{-1} = \mathbf{y}/\langle \mathbf{y},\mathbf{u}\rangle \in E^*$ and $\mathbf{y}^{-1} = \mathbf{u}/\langle \mathbf{y},\mathbf{u}\rangle \in E$ [3]. Keeping this definition in mind, we see that the inverse used in the first recurrence relation of the algorithm is the inverse of $\widehat{\varepsilon}_{2k}^{(n+1)} - \widehat{\varepsilon}_{2k}^{(n)}$ in the couple $(\widehat{\varepsilon}_{2k}^{(n+1)} - \widehat{\varepsilon}_{2k}^{(n)}, \mathbf{y}) \in E \times E^*$, while, in the second relation, we use the inverse of $\widehat{\varepsilon}_{2k+1}^{(n+1)} - \widehat{\varepsilon}_{2k+1}^{(n)}$ in the couple $(\widehat{\varepsilon}_{2k}^{(n+1)} - \widehat{\varepsilon}_{2k}^{(n)}, \widehat{\varepsilon}_{2k+1}^{(n+1)} - \widehat{\varepsilon}_{2k+1}^{(n)}) \in E \times E^*$.

The elements $\widetilde{\mathbf{e}}_{\mathbf{k}}(\mathbf{S}_n) \in E$ of the second topological Shanks transformation can be recursively computed by the *second topological ε-algorithm* (TEA2) [3], whose rules are (see [9] for a FORTRAN subroutine)

$$(4.2)\quad \begin{cases} \widetilde{\varepsilon}_{-1}^{(n)} &= \mathbf{0} \in E^*, \qquad n = 0,1,\ldots, \\ \widetilde{\varepsilon}_{0}^{(n)} &= \mathbf{S}_n \in E, \qquad n = 0,1,\ldots, \\ \widetilde{\varepsilon}_{2k+1}^{(n)} &= \widetilde{\varepsilon}_{2k-1}^{(n+1)} + \dfrac{\mathbf{y}}{\langle \mathbf{y}, \widetilde{\varepsilon}_{2k}^{(n+1)} - \widetilde{\varepsilon}_{2k}^{(n)}\rangle} \in E^*, \quad k,n = 0,1,\ldots, \\ \widetilde{\varepsilon}_{2k+2}^{(n)} &= \widetilde{\varepsilon}_{2k}^{(n+1)} + \dfrac{\widetilde{\varepsilon}_{2k}^{(n+2)} - \widetilde{\varepsilon}_{2k}^{(n+1)}}{\langle \widetilde{\varepsilon}_{2k+1}^{(n+1)} - \widetilde{\varepsilon}_{2k+1}^{(n)}, \widetilde{\varepsilon}_{2k}^{(n+2)} - \widetilde{\varepsilon}_{2k}^{(n+1)}\rangle} \in E, \quad k,n = 0,1,\ldots. \end{cases}$$

The following property holds.

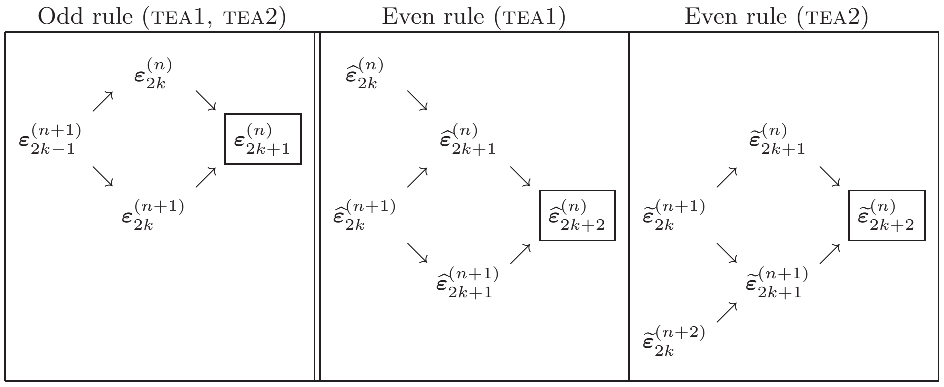
Odd rule (TEA1, TEA2)
Even rule (TEA1)
Even rule (TEA2)
$\boldsymbol{\varepsilon}_{2k}^{(n)}$
$\boldsymbol{\varepsilon}_{2k-1}^{(n+1)}$
$\boldsymbol{\varepsilon}_{2k+1}^{(n)}$
$\boldsymbol{\varepsilon}_{2k}^{(n+1)}$
$\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}$
$\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)}$
$\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}$
$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)}$
$\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n+1)}$
$\widetilde{\boldsymbol{\varepsilon}}_{2k+1}^{(n)}$
$\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}$
$\widetilde{\boldsymbol{\varepsilon}}_{2k+2}^{(n)}$
$\widetilde{\boldsymbol{\varepsilon}}_{2k+1}^{(n+1)}$
$\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+2)}$

FIG. 2. *The relations for the first (*TEA1*) and the second (*TEA2*) topological ε-algorithms (in the odd rule $\boldsymbol{\varepsilon}$ is $\widehat{\boldsymbol{\varepsilon}}$ for* TEA1 *and $\widetilde{\boldsymbol{\varepsilon}}$ for* TEA2*).*

PROPERTY 4.1 (see [3]).

$$\begin{aligned} &\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} = \widehat{\mathbf{e}}_k(\mathbf{S}_n), && \langle \mathbf{y}, \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}\rangle = e_k(\langle \mathbf{y}, \mathbf{S}_n\rangle), \\ &\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)} = \mathbf{y}/\langle \mathbf{y}, \widehat{\mathbf{e}}_k(\Delta \mathbf{S}_n)\rangle, && \widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)} = \mathbf{y}/e_k(\langle \mathbf{y}, \Delta \mathbf{S}_n\rangle), \qquad k, n = 0, 1, \ldots. \end{aligned}$$

These relations are also true for the $\widetilde{\boldsymbol{\varepsilon}}_k^{(n)}$'s and the $\widetilde{\mathbf{e}}_k$'s.

Remark 4.2. Due to the first rule of the algorithms, $\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)} = \mathbf{y}\sum_{i=0}^{k} 1/\langle \mathbf{y}, \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}\rangle$, and a similar relation for $\widetilde{\boldsymbol{\varepsilon}}_{2k+1}^{(n)}$, which shows that these elements of E^* are multiples of $\mathbf{y}$.

If $E = \mathbb{R}$ or $\mathbb{C}$, the rules (4.1) and (4.2) of the first and the second topological ε-algorithms reduce to those of the scalar ε-algorithm.

The relationships between the elements of E and E^* in the topological ε-array involved in these two topological algorithms are shown in Figure 2. The odd rules (those for computing elements with an odd lower index) have the same structure as the rule of the scalar ε-algorithm, but the even ones need an additional term of the topological ε-array.

In the first and the second topological ε-algorithms, the difficulty could be to compute the duality products appearing in their denominators since they involve linear functionals which, in the general case, cannot be easily handled on a computer when E is a general vector space (for example, if $\mathbf{S}_n$ is a function and if $\langle \mathbf{y}, \mathbf{S}_n\rangle = \int_a^b \mathbf{S}_n(t)dt$). Of course, when $E = \mathbb{R}^m$ or $\mathbb{R}^{m\times s}$, handling linear functionals is trivial because those spaces are their own dual spaces. However, the duality product has to be used recursively at each step of the algorithms. Moreover, in the even rules of the algorithms, the duality has to be taken with linear functionals computed recursively by the odd rules. These computational problems will be solved by the new algorithms given in the next section.

5. The simplified topological $\boldsymbol{\varepsilon}$-algorithms.

We will now derive new algorithms for implementing the two topological Shanks sequence transformations of section 3 which avoid the manipulation of elements of E^* and the use of the duality product with $\mathbf{y}$ inside the rules of the algorithms. In these new algorithms, the linear functional $\mathbf{y}$ will only be applied to the terms of the initial sequence $(\mathbf{S}_n)$. Moreover, they require the storage of a fewer number of elements of E and no element of E^*

since they connect only terms with an even lower index in the topological ε-array. Their implementation will be described in section 7.

5.1. The first simplified topological ε-algorithm. Applying Wynn's scalar ε-algorithm (2.4) to the sequence $(S_n = \langle \mathbf{y}, \mathbf{S}_n\rangle)$, from Property 4.1, we have $\varepsilon_{2k}^{(n)} = e_k(\langle \mathbf{y}, \mathbf{S}_n\rangle)$ and, by Property 2.2, we obtain

$$\begin{aligned}\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)} &= \frac{\mathbf{y}}{\langle \mathbf{y}, \widehat{\mathbf{e}}_k(\Delta \mathbf{S}_{n+1})\rangle} - \frac{\mathbf{y}}{\langle \mathbf{y}, \widehat{\mathbf{e}}_k(\Delta \mathbf{S}_{n})\rangle}\\ &= \mathbf{y}\left(\frac{1}{e_k(\langle \mathbf{y}, \Delta\mathbf{S}_{n+1}\rangle)} - \frac{1}{e_k(\langle \mathbf{y}, \Delta\mathbf{S}_{n}\rangle)}\right)\\ &= \mathbf{y}(\varepsilon_{2k+1}^{(n+1)} - \varepsilon_{2k+1}^{(n)}).\end{aligned}$$

Thus, the last relation in (4.1) becomes

$$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} = \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} + \frac{\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}}{\langle \mathbf{y}, \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}\rangle(\varepsilon_{2k+1}^{(n+1)} - \varepsilon_{2k+1}^{(n)})}, \qquad k, n = 0, 1, \ldots.$$

Moreover, due to Property 4.1, the quantities $\langle \mathbf{y}, \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}\rangle$ can be computed by the scalar ε-algorithm, and we finally obtain the rule

$$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} = \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} + \frac{1}{(\varepsilon_{2k}^{(n+1)} - \varepsilon_{2k}^{(n)})(\varepsilon_{2k+1}^{(n+1)} - \varepsilon_{2k+1}^{(n)})}(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}), \; k, n = 0, 1, \ldots, \tag{5.1}$$

with $\widehat{\boldsymbol{\varepsilon}}_0^{(n)} = \mathbf{S}_n \in E$, $n = 0, 1, \ldots$.

This relation was already given in [3] as Property 10 but not in an algorithmic form. It shows that the first topological Shanks transformation can now be implemented by a triangular scheme involving the scalar ε-algorithm and elements of E exclusively, instead of extended rhombus rules needing the duality product with elements of E^*. Moreover, we now have only one rule instead of two.

Thanks to the recursive rule of the scalar ε-algorithm, the algorithm (5.1) can also be written under one of the following equivalent forms:

$$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} = \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} + \frac{\varepsilon_{2k+1}^{(n)} - \varepsilon_{2k-1}^{(n+1)}}{\varepsilon_{2k+1}^{(n+1)} - \varepsilon_{2k+1}^{(n)}}(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}), \tag{5.2}$$

$$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} = \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} + \frac{\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n+1)}}{\varepsilon_{2k}^{(n+1)} - \varepsilon_{2k}^{(n)}}(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}), \tag{5.3}$$

$$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} = \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} + (\varepsilon_{2k+1}^{(n)} - \varepsilon_{2k-1}^{(n+1)})(\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n+1)})(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}). \tag{5.4}$$

Notice that (5.3) can also be written as

$$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} = \frac{\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n)}}{\varepsilon_{2k}^{(n+1)} - \varepsilon_{2k}^{(n)}}\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \frac{\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n+1)}}{\varepsilon_{2k}^{(n+1)} - \varepsilon_{2k}^{(n)}}\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} \tag{5.5}$$

$$= \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} + \frac{\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n)}}{\varepsilon_{2k}^{(n+1)} - \varepsilon_{2k}^{(n)}}(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}). \tag{5.6}$$

This new algorithm, in any of the preceding forms, is called the *first simplified topological ε-algorithm*, and it is denoted by STEA1.

Taking the duality product of $\mathbf{y}$ with any of the preceding relations (5.1)–(5.6) allows one to recover the rule of the scalar ε-algorithm or leads to an identity. The same is true when E is $\mathbb{R}$ or $\mathbb{C}$.

From (5.5), we have the following property.

PROPERTY 5.1. *The computation of $\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)}$ is stable for all n if $\exists M_k$, independent of n, such that*

$$\left|\frac{\varepsilon_{2k+2}^{(n)}-\varepsilon_{2k}^{(n)}}{\varepsilon_{2k}^{(n+1)}-\varepsilon_{2k}^{(n)}}\right|+\left|\frac{\varepsilon_{2k+2}^{(n)}-\varepsilon_{2k}^{(n+1)}}{\varepsilon_{2k}^{(n+1)}-\varepsilon_{2k}^{(n)}}\right|\le M_k.$$

Remark 5.2. Although the $\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)}$'s are no longer needed, it holds from Property 4.1 that $\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)}=\widehat{\boldsymbol{\varepsilon}}_{2k-1}^{(n+1)}+\mathbf{y}/(\varepsilon_{2k}^{(n+1)}-\varepsilon_{2k}^{(n)})$, and thus, similarly to Property 2.3, we obtain $\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)}=\mathbf{y}\sum_{i=0}^{k}1/\Delta\varepsilon_{2i}^{(n+k-i)}$. Since $\Delta S_n=\langle\mathbf{y},\Delta\mathbf{S}_n\rangle$, we also have, from what precedes and Property 4.1, $\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)}=\mathbf{y}\varepsilon_{2k+1}^{(n)}$ and $\varepsilon_{2k+1}^{(n)}=1/e_k(\langle\mathbf{y},\Delta\mathbf{S}_n\rangle)$. These relations also hold for the $\widetilde{\boldsymbol{\varepsilon}}_{2k+2}^{(n)}$'s of the second simplified topological ε-algorithm (see below).

5.2. The second simplified topological ε-algorithm. Since similar algebraic properties hold for the second topological Shanks transformation and the second topological ε-algorithm, we can derive, in the same way, a *second simplified topological ε-algorithm*, denoted by STEA2. Its rules can be written in any of the four following equivalent forms:

$$\widetilde{\boldsymbol{\varepsilon}}_{2k+2}^{(n)}=\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}+\frac{1}{(\varepsilon_{2k}^{(n+2)}-\varepsilon_{2k}^{(n+1)})(\varepsilon_{2k+1}^{(n+1)}-\varepsilon_{2k+1}^{(n)})}(\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+2)}-\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}),$$

$$\widetilde{\boldsymbol{\varepsilon}}_{2k+2}^{(n)}=\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}+\frac{\varepsilon_{2k+1}^{(n+1)}-\varepsilon_{2k-1}^{(n+2)}}{\varepsilon_{2k+1}^{(n+1)}-\varepsilon_{2k+1}^{(n)}}(\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+2)}-\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}),$$

$$\widetilde{\boldsymbol{\varepsilon}}_{2k+2}^{(n)}=\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}+\frac{\varepsilon_{2k+2}^{(n)}-\varepsilon_{2k}^{(n+1)}}{\varepsilon_{2k}^{(n+2)}-\varepsilon_{2k}^{(n+1)}}(\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+2)}-\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}),$$

$$\widetilde{\boldsymbol{\varepsilon}}_{2k+2}{}^{(n)}=\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}+(\varepsilon_{2k+1}^{(n+1)}-\varepsilon_{2k-1}^{(n+2)})(\varepsilon_{2k+2}^{(n)}-\varepsilon_{2k}^{(n+1)})(\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+2)}-\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+1)}),$$

with $\widetilde{\boldsymbol{\varepsilon}}_{0}^{(n)}=\mathbf{S}_n\in E$, $n=0,1,\ldots$.

In these formulae, the scalar quantities are the same as those used in the first simplified topological ε-algorithm, and they are again obtained by applying the scalar ε-algorithm to the sequence $(\langle\mathbf{y},\mathbf{S}_n\rangle)$.

In Figure 3, we show the very simple triangular relationships between the elements of E in the topological ε-array for the two simplified algorithms. Remark that only the even terms are needed and computed.

Remark 5.3. Since the $\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}$ and the $\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n)}$ are related by a triangular recursive scheme, they satisfy the theory of *reference functionals* developed in [12].

6. Convergence and acceleration. Several convergence and acceleration results for the topological ε-algorithm were already given in [3]. However, the conditions of these theorems are difficult to check since they involved elements of E^* which are recursively computed. The fact that now the simplified algorithms no longer involve these linear functionals will allow us to obtain new results, easier to verify in practice. Thus, the simplified ε-algorithms not only play a major role in the implementation of

STEA1

$$\begin{tikzcd}
\widehat{\varepsilon}^{(n)}_{2k} \arrow[dr] & \\
\widehat{\varepsilon}^{(n+1)}_{2k} \arrow[r] & \boxed{\widehat{\varepsilon}^{(n)}_{2k+2}}
\end{tikzcd}$$

STEA2

$$\begin{tikzcd}
\widetilde{\varepsilon}^{(n+1)}_{2k} \arrow[r] & \boxed{\widetilde{\varepsilon}^{(n)}_{2k+2}} \\
\widetilde{\varepsilon}^{(n+2)}_{2k} \arrow[ur] &
\end{tikzcd}$$

FIG. 3. *The relations for the first* (STEA1) *and the second* (STEA2) *simplified topological ε-algorithms.*

the topological Shanks transformations, but they also have a primary impact on the theoretical results that can be proved.

Property 3.1 means that the numbers $e_k(S_n)$, obtained by applying the scalar Shanks transformation to the sequence of numbers $(S_n = \langle \mathbf{y}, \mathbf{S}_n \rangle)$, and the elements $\widehat{\mathbf{e}}_k(\mathbf{S}_n)$ and $\widetilde{\mathbf{e}}_k(\mathbf{S}_n)$ of E, obtained by the topological Shanks transformations applied to the sequence $(\mathbf{S}_n)$ of elements of E, are computed by a linear combination with the same coefficients. It implies that, under some circumstances, the sequences $(e_k(S_n))$, $(\widehat{\mathbf{e}}_k(\mathbf{S}_n))$, and $(\widetilde{\mathbf{e}}_k(\mathbf{S}_n))$ can share similar convergence and acceleration behaviors, and we will now illustrate this fact.

Let us begin by two results about important classes of sequences which, despite their simplicity, remained unnoticed.

DEFINITION 6.1. *A sequence of vectors in $\mathbb{R}^m$ or matrices in $\mathbb{R}^{m\times s}$ is* totally monotonic*, and we write $(\mathbf{S}_n) \in TM$ if, for all k and n, $(-1)^k \Delta^k \mathbf{S}_n \geq \mathbf{0}$, where the inequality has to be understood for each component of the vectors or each element of the matrices. A sequence of vectors or matrices is* totally oscillating*, and we write $(\mathbf{S}_n) \in TO$ if $((-1)^n \mathbf{S}_n) \in TM$.*

In the scalar case, these sequences were first studied by Wynn [41], and his results were complemented by Brezinski [1, 2]. Their construction and their properties were studied in [5, 42]. For the vector and the matrix cases, we have the following new results.

THEOREM 6.2. *If $(\mathbf{S}_n)$ converges to $\mathbf{S}$, if $(S_n = \langle \mathbf{y}, \mathbf{S}_n \rangle) \in TM$, and if $\exists a \neq 0$ and $\mathbf{b} \in \mathbb{R}^m$ (or $\mathbb{R}^{m\times s}$ in the matrix case) such that $(a\mathbf{S}_n + \mathbf{b}) \in TM$, then*

$$\begin{aligned}
&\mathbf{0} \leq a\widehat{\varepsilon}^{(n)}_{2k+2} + \mathbf{b} \leq a\widehat{\varepsilon}^{(n)}_{2k} + \mathbf{b}, && \mathbf{0} \leq a\widehat{\varepsilon}^{(n+1)}_{2k} + \mathbf{b} \leq a\widehat{\varepsilon}^{(n)}_{2k} + \mathbf{b},\\
&\mathbf{0} \leq a\widehat{\varepsilon}^{(n)}_{2k+2} + \mathbf{b} \leq a\widehat{\varepsilon}^{(n+1)}_{2k} + \mathbf{b}, && \mathbf{0} \leq a\widehat{\varepsilon}^{(n)}_{2k+2} + \mathbf{b} \leq a\widehat{\varepsilon}^{(n+2)}_{2k} + \mathbf{b},\\
&\forall k,\ \textit{fixed},\quad \lim_{n\to\infty} \widehat{\varepsilon}^{(n)}_{2k} = \mathbf{S}, && \forall n,\ \textit{fixed},\quad \lim_{k\to\infty} \widehat{\varepsilon}^{(n)}_{2k} = \mathbf{S}.
\end{aligned}$$

Moreover, if $\lim_{n\to\infty} \langle \mathbf{y}, \mathbf{S}_{n+1} - \mathbf{S}\rangle / \langle \mathbf{y}, \mathbf{S}_n - \mathbf{S}\rangle \neq 1$, then

$$\forall k,\ \textit{fixed},\quad \lim_{n\to\infty} \|\widehat{\varepsilon}^{(n)}_{2k} - \mathbf{S}\| / \|\mathbf{S}_{n+2k} - \mathbf{S}\| = 0,$$

$$\forall n,\ \textit{fixed},\quad \lim_{k\to\infty} \|\widehat{\varepsilon}^{(n)}_{2k} - \mathbf{S}\| / \|\mathbf{S}_{n+2k} - \mathbf{S}\| = 0.$$

The same results hold for the $\widetilde{\varepsilon}^{(n)}_{2k}$.

Proof. Since the coefficients $a_i^{(n,k)}$ in (2.3) and (3.3) are the same, then, if $(\mathbf{S}_n) \in$ TM or TO and if $(S_n = \langle \mathbf{y}, \mathbf{S}_n \rangle) \in$ TM or TO, the scalars $\varepsilon^{(n)}_{2k}$, the components of the vectors or the elements of the matrices $\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}$, and those of $\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n)}$ satisfy the same inequalities, which proves the results. $\square$

Equivalent results hold for TO sequences of vectors and matrices. The proof is the same as for the preceding theorem.

THEOREM 6.3. *If* $(\mathbf{S}_n)$ *converges to* $\mathbf{S}$*, if* $(S_n = \langle \mathbf{y}, \mathbf{S}_n\rangle) \in$ *TO, and if* $\exists a \neq 0$ *and* $\mathbf{b} \in \mathbb{R}^m$ *(or* $\mathbb{R}^{m\times s}$ *in the matrix case) such that* $(a\mathbf{S}_n + \mathbf{b}) \in$ *TO, then*

$$\begin{array}{ll}
\mathbf{0} \le a\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(2n)} + \mathbf{b} \le a\widehat{\boldsymbol{\varepsilon}}_{2k}^{(2n)} + \mathbf{b}, & a(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(2n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(2n)}) \le a(\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(2n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(2n)}) \le \mathbf{0}, \\
a\widehat{\boldsymbol{\varepsilon}}_{2k}^{(2n+1)} + \mathbf{b} \le a\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(2n+1)} + \mathbf{b} \le \mathbf{0}, & \mathbf{0} \le a(\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(2n+2)} - \widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(2n+1)}) \le a(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(2n+2)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(2n+1)}), \\
\mathbf{0} \le a\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(2n)} + \mathbf{b} \le a\widehat{\boldsymbol{\varepsilon}}_{2k}^{(2n+2)} + \mathbf{b}, & \forall k,\ \textit{fixed}, \quad \lim_{n\to\infty} \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} = \mathbf{S}, \\
a\widehat{\boldsymbol{\varepsilon}}_{2k}^{(2n+3)} + \mathbf{b} \le a\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(2n+1)} + \mathbf{b} \le \mathbf{0}, & \forall n,\ \textit{fixed}, \quad \lim_{k\to\infty} \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} = \mathbf{S}.
\end{array}$$

Moreover,

$$\forall k,\ \textit{fixed}, \quad \lim_{n\to\infty} \|\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S}\| / \|\mathbf{S}_{n+2k} - \mathbf{S}\| = 0,$$
$$\forall n,\ \textit{fixed}, \quad \lim_{k\to\infty} \|\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S}\| / \|\mathbf{S}_{n+2k} - \mathbf{S}\| = 0.$$

The same results hold for the $\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n)}$*.*

Remark 6.4. If the vectors $\mathbf{S}_n$ belong to TM or TO and if $\mathbf{y} \ge \mathbf{0}$, then $(S_n = \langle \mathbf{y}, \mathbf{S}_n\rangle) \in$ TM or TO. If the matrices $\mathbf{S}_n$ belong to TM and if $\mathbf{y}$ is the linear form such that, for any matrix $\mathbf{M}$, $\langle \mathbf{y}, \mathbf{M}\rangle = \mathrm{trace}(\mathbf{M})$, then $(S_n = \langle \mathbf{y}, \mathbf{S}_n\rangle) \in$ TM or TO. Thus, the convergence of the first simplified topological ε-algorithm depends on the behavior of the scalar one, and the choice of $\mathbf{y}$ intervenes in the conditions to be satisfied. As noticed in [25], if, for all n, $\mathbf{S}_n \in \mathbb{R}^m$ is replaced by $\mathbf{D}\mathbf{S}_n$, where $\mathbf{D}$ is a diagonal regular matrix of dimension m, and if $\mathbf{y}$ is replaced by $\lambda\mathbf{D}^{-1}\mathbf{y}$, where λ is a nonzero scalar, then, for all k and n, $\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}$ becomes $\mathbf{D}\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}$ and $\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)}$ becomes $\mathbf{D}^{-1}\widehat{\boldsymbol{\varepsilon}}_{2k+1}^{(n)}$. The same property holds for the $\widetilde{\boldsymbol{\varepsilon}}_{k}^{(n)}$'s. This remark allows one to extend the two previous theorems.

Let us continue by some other convergence and acceleration results that would not have been easily obtained directly from the determinantal formulae of the topological Shanks transformations or from the rules of the topological ε-algorithms.

Assume now that E is a normed vector space and that $\mathbf{y} \in E' \subseteq E^*$, where E' is the space of all continuous linear functionals on E. We will consider only the case of the first transformation and the first algorithm; the second ones could be treated similarly and lead to equivalent results.

Let us set

$$r_k^{(n)} = (\varepsilon_{2k+2}^{(n)} - \varepsilon_{2k}^{(n+1)})/(\varepsilon_{2k}^{(n+1)} - \varepsilon_{2k}^{(n)}). \tag{6.1}$$

The rule (5.3) of the first simplified topological ε-algorithm gives

$$\left\| \widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} \right\| / \left\| \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} \right\| = |r_k^{(n)}|,$$

which shows that both ratios have the same form and behave similarly. We have the following convergence result.

THEOREM 6.5. *If* $\lim_{n\to\infty} \widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} = \mathbf{S}$*, and if* $\exists M$ *such that, for all* $n \ge N$*,* $|r_k^{(n)}| \le M$*, then* $\lim_{n\to\infty} \widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} = \mathbf{S}$*.*

Proof. The relation (5.3) can be written as

$$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} = (1 + r_k^{(n)})\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - r_k^{(n)}\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}.$$

By the assumption on $r_k^{(n)}$ and since the two scalar coefficients in the right-hand side sum up to 1, the conditions of the Toeplitz theorem for summation processes are satisfied, which proves the result. ☐

The other forms of the first simplified ε-algorithm lead to different ratios linking its behavior with that of the scalar ε-algorithm

Let us now give an acceleration result in the case where $E = \mathbb{R}^m$, $m > 1$.

THEOREM 6.6. *Assume that $E = \mathbb{R}^m$, that $r_k^{(n)} = r_k + o(1)$ with $r_k \neq -1$, and that for all i, $(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \mathbf{S})_i/(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S})_i = r_k/(1 + r_k) + o(1)$. Then $\lim_{n\to\infty} \|\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} - \mathbf{S}\|/\|\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S}\| = 0$.*

Proof. Writing (5.3) as (which is the same as (5.5))

$$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} - \mathbf{S} = (1 + r_k^{(n)})(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n+1)} - \mathbf{S}) - r_k^{(n)}(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S}),$$

we have from the assumptions on $r_k^{(n)}$ and on the components of the vectors

$$(\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} - \mathbf{S})_i = [(1+r_k+o(1))(r_k/(1+r_k)+o(1)) - (r_k+o(1))](\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S})_i = o(1)(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S})_i.$$

Thus, we prove the result since all norms are equivalent. ☐

Remark 6.7. By the assumptions on the components of the vectors, it holds, for the scalar ε-algorithm, that $\lim_{n\to\infty}(\varepsilon_{2k}^{(n+1)} - S)/(\varepsilon_{2k}^{(n)} - S) = r_k/(1 + r_k)$. Since $r_k^{(n)} = r_k + o(1)$ and

$$r_k^{(n)} = \left(\frac{\varepsilon_{2k+2}^{(n)} - S}{\varepsilon_{2k}^{(n)} - S} - \frac{\varepsilon_{2k}^{(n+1)} - S}{\varepsilon_{2k}^{(n)} - S}\right) \Big/ \left(\frac{\varepsilon_{2k}^{(n+1)} - S}{\varepsilon_{2k}^{(n)} - S} - 1\right),$$

it follows that we also have $\lim_{n\to\infty}(\varepsilon_{2k+2}^{(n)} - S)/(\varepsilon_{2k}^{(n)} - S) = 0$.

Let us now give convergence and acceleration results concerning four special types of sequences. These sequences were considered by Wynn in the scalar case for the scalar ε-algorithm [41]. We will see that his results can be extended to similar sequences of elements of E. The first result is the following.

THEOREM 6.8. *We consider sequences of the form*

$$\text{(a) } \mathbf{S_n} - \mathbf{S} \sim \sum_{\mathrm{i}=1}^{\infty} \mathrm{a_i}\lambda_\mathrm{i}^\mathrm{n}\mathbf{u_i} \quad (\mathrm{n} \to \infty) \quad \textit{or} \quad \text{(b) } \mathbf{S_n} - \mathbf{S} \sim (-1)^\mathrm{n}\sum_{\mathrm{i}=1}^{\infty} \mathrm{a_i}\lambda_\mathrm{i}^\mathrm{n}\mathbf{u_i} \quad (\mathrm{n} \to \infty),$$

where $a_i, \lambda_i \in \mathbb{K}$, $\mathbf{u_i} \in E$, and $1 > \lambda_1 > \lambda_2 > \cdots > 0$. Then, when k is fixed and n tends to infinity,

$$\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S} = \mathcal{O}(\lambda_{k+1}^n), \qquad \frac{\|\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} - \mathbf{S}\|}{\|\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S}\|} = \mathcal{O}((\lambda_{k+2}/\lambda_{k+1})^n).$$

Proof. We consider sequences of the form (a). We have

$$\langle \mathbf{y}, \mathbf{S}_n \rangle - \langle \mathbf{y}, \mathbf{S} \rangle \sim \sum_{i=1}^{\infty} a_i\lambda_i^n \langle \mathbf{y}, \mathbf{u_i} \rangle.$$

If the scalar ε-algorithm is applied to the sequence $(\langle \mathbf{y}, \mathbf{S}_n\rangle)$, Wynn [41] proved that

$$\varepsilon_{2k}^{(n)} - \langle \mathbf{y}, \mathbf{S}\rangle = a_{k+1}\frac{(\lambda_{k+1}-\lambda_1)^2\cdots(\lambda_{k+1}-\lambda_k)^2}{(1-\lambda_1)^2\cdots(1-\lambda_k)^2}\lambda_{k+1}^n\langle \mathbf{y}, \mathbf{u_{k+1}}\rangle + \mathcal{O}(\lambda_{k+2}^n).$$

After some algebraic manipulations using (6.1), we get

$$r_k^{(n)} \sim \frac{\lambda_{k+1}}{1-\lambda_{k+1}} + \frac{a_{k+2}\langle \mathbf{y}, \mathbf{u}_{k+2}\rangle(\lambda_{k+2}-\lambda_1)^2\cdots(\lambda_{k+2}-\lambda_{k+1})^2}{a_{k+1}\langle \mathbf{y}, \mathbf{u}_{k+1}\rangle(\lambda_{k+1}-\lambda_1)^2\cdots(\lambda_{k+1}-\lambda_k)^2(1-\lambda_{k+1})^3}\left(\frac{\lambda_{k+2}}{\lambda_{k+1}}\right)^n + o\left(\frac{\lambda_{k+2}}{\lambda_{k+1}}\right)^n.$$

Assume that

$$\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S} \sim \sum_{i=k+1}^{p} \alpha_{k,i}\lambda_i^n \mathbf{u}_i,$$

where the coefficients $\alpha_{k,i}$ depend on k and i but not on n (by assumption, this is true for $k = 0$).

Plugging this relation into the rule (5.3) of the first simplified topological ε-algorithm, and using the preceding expression for $r_k^{(n)}$, we obtain (see [4] for a similar proof) that

$$\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} - \mathbf{S} \sim \frac{1}{1-\lambda_{k+1}}\left(\sum_{i=k+1}^{p}\alpha_{k,i}\lambda_i^{n+1}\mathbf{u}_i - \lambda_{k+1}\sum_{i=k+1}^{p}\alpha_{k,i}\lambda_i^n\mathbf{u}_i\right),$$

which proves the first result by induction. The second result follows immediately.

The proof for sequences of the form (b) is similar to the preceding case by replacing each λ_i by $-\lambda_i$. ☐

An application of this result will be given in section 8.1.

The next theorem is obtained by expressing $r_k^{(n)}$ as above and using the result of Wynn for the scalar case [41].

THEOREM 6.9. *We consider sequences of the form*

$$\mathbf{S_n} - \mathbf{S} \sim \sum_{i=1}^{\infty} a_i(n+b)^{-i}\mathbf{u_i} \quad (n \to \infty),$$

where $a_i, b \in \mathbb{K}$, $\mathbf{u_i} \in E$. *Then, when k is fixed and n tends to infinity,*

$$\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S} \sim \frac{a_1\mathbf{u_1}}{(k+1)(n+b)}.$$

Thus, when n tends to infinity, the sequence $(\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)})$ does not converge to $\mathbf{S}$ faster than the sequence $(\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)})$. We finally have a last result whose proof uses (5.6) and the expression given by Wynn [41] in the scalar case.

THEOREM 6.10. *We consider sequences of the form*

$$\mathbf{S_n} - \mathbf{S} \sim (-1)^n\sum_{i=1}^{\infty} a_i(n+b)^{-i}\mathbf{u_i} \quad (n \to \infty),$$

where $a_i, b \in \mathbb{K}$, $\mathbf{u_i} \in E$. Then, when k is fixed and n tends to infinity,

$$\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S} \sim (-1)^n \frac{a_1(k\,!)^2\mathbf{u_1}}{4^k(n+b)^{2k+1}}, \qquad \frac{\|\widehat{\boldsymbol{\varepsilon}}_{2k+2}^{(n)} - \mathbf{S}\|}{\|\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)} - \mathbf{S}\|} = \mathcal{O}(1/(n+b)).$$

The results of [16] about sequences with more complicated asymptotic expansions could possibly be extended similarly to the first simplified topological ε-algorithm. Since the vector E-algorithm [7] can be used for implementing the first topological Shanks transformation, the acceleration results proved for it should also be valid for the first simplified topological ε-algorithm [23].

7. Implementation and performances.

To implement all the ε-algorithms and to construct the ε-array, the simplest procedure is to store all its elements for all k and n. Starting from a given number of terms in the two initial columns (the second one for the simplified algorithms), the other columns are computed one by one, each of them having one less term than the preceding one (two less terms for the simplified algorithms). Thus, a triangular part of the ε-array is obtained (only the even columns in the simplified cases), as can be seen in Figure 1. However, such a procedure requires storing all the elements of this triangular part, and it can be costly for sequences of vectors or matrices.

A better procedure, which avoids storing the whole triangular array, is to add each new term of the original sequence one by one and to proceed with the computation of the ε-array by ascending diagonal as far as possible (or required). This technique was invented by Wynn for the scalar ε-algorithm and other algorithms having a similar structure [38, 40] (see also [9, pp. 397ff.]), including the vector ε-algorithm [37]. It can be summarized as follows: after having computed a triangular part of the ε-array and stored only its last ascending diagonal (for example, the ascending diagonal $\varepsilon_0^{(3)}, \varepsilon_1^{(2)}, \varepsilon_2^{(1)}, \varepsilon_3^{(0)}$ of Figure 1 which contains all the terms needed for computing the next diagonal), a new element of the initial sequence is introduced ($\varepsilon_0^{(4)} = S_4$ in our example), and the next ascending diagonal is computed element by element, that is, $\varepsilon_1^{(3)}, \varepsilon_2^{(2)}, \varepsilon_3^{(1)}, \varepsilon_4^{(0)}$, by using the rhombus rule of the algorithm. This technique requires only the storage of one ascending diagonal and three temporary auxiliary elements. Let us mention that, for computing the descending diagonal $\varepsilon_0^{(0)}, \varepsilon_1^{(0)}, \varepsilon_2^{(0)}, \ldots$, the new ascending diagonal has to be computed as far as possible, while, for computing the descending column $\varepsilon_{2k}^{(0)}, \varepsilon_{2k}^{(1)}, \varepsilon_{2k}^{(2)}, \ldots$, for a fixed value of k, the new ascending diagonal has to be computed only until the column $2k$ has been reached.

In the topological ε-algorithms, the odd rule has the same form as the rule of the scalar ε-algorithm, but the even rule needs an extra element, as shown in Figure 2. This is not a problem for implementing TEA2 by exactly the same technique as Wynn's since the extra term, namely $\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(n+2)}$ needed for computing $\widetilde{\boldsymbol{\varepsilon}}_{2k+2}^{(n)}$, was already computed in the diagonal we are constructing. Thus, it is necessary to store only the preceding diagonal consisting of elements of E and E^*.

Contrarily to TEA2, for implementing TEA1, the extra element, namely $\widehat{\boldsymbol{\varepsilon}}_{2k}^{(n)}$, is in the diagonal above the preceding one. Thus, in addition to the full preceding diagonal, we also need to store the even elements of the diagonal above it. Thus, in total, one and a half diagonals (of elements of E and E^*) have to be stored.

For the simplified topological ε-algorithms, only elements with an even lower index are used and computed, and the rules are shown in Figure 3. Similar considerations as above can be made for the new algorithms. Thus, for STEA1, by the ascending diagonal technique, one need only store the elements of E, that is, those with an even lower index, located in the two preceding diagonals, and, for STEA2, only the elements of E located in the preceding diagonal.

Of course, we also have to compute, by the ascending diagonal technique of Wynn, the elements of the scalar ε-array, but this is cheep in terms of storage requirements and arithmetical operations. Each new scalar term S_n is obtained by computing the duality product of $\mathbf{S}_n$ with $\mathbf{y}$ and is immediately used for building the new ascending diagonal of the scalar ε-array. Then, the new ascending diagonal of the topological ε-array consisting only of elements of E is constructed. A paper with the corresponding MATLAB programs is in preparation.

The storage requirements of the four topological algorithms for computing $\widehat{\boldsymbol{\varepsilon}}_{2k}^{(0)}$ or $\widetilde{\boldsymbol{\varepsilon}}_{2k}^{(0)}$, including the temporary auxiliary elements, are given in Table 1.

Table 1
Storage requirements.

Algorithm	# elements $\boldsymbol{\varepsilon}$-array	In spaces	# auxiliary elements
TEA1	$3k$	E, E^*	3
STEA1	$2k$	E	2
TEA2	$2k$	E, E^*	3
STEA2	k	E	2

Let us now discuss some possible choices for the linear functional $\mathbf{y} \in E^*$. When $E = \mathbb{C}^m$, we can define it as $\mathbf{y} : \mathbf{S}_n \in \mathbb{C}^m \longmapsto \langle \mathbf{y}, \mathbf{S}_n \rangle = (\mathbf{y}, \mathbf{S}_n)$, the usual inner product of the vectors $\mathbf{y}$ and $\mathbf{S}_n$, or, more generally, as $(\mathbf{y}, \mathbf{M}\mathbf{S}_n)$, where $\mathbf{M}$ is some matrix.

When $E = \mathbb{C}^{m\times m}$, the linear functional $\mathbf{y} \in E^*$ can be defined as $\mathbf{y} : \mathbf{S}_n \in \mathbb{C}^{m\times m} \longmapsto \langle \mathbf{y}, \mathbf{S}_n \rangle = \mathrm{trace}(\mathbf{S}_n)$ or, more generally, for $\mathbf{S}_n \in \mathbb{C}^{m\times s}$, as $\mathrm{trace}(\mathbf{Y}^T\mathbf{S}_n)$, where $\mathbf{Y} \in \mathbb{C}^{m\times s}$. We can also define $\mathbf{y}$ by $(\mathbf{u}, \mathbf{S}_n\mathbf{v})$, where $\mathbf{u} \in \mathbb{C}^m$ and $\mathbf{v} \in \mathbb{C}^s$.

The simplified topological ε-algorithms for implementing the topological Shanks transformations also allow us to improve its numerical stability. This is due to the existence, for the scalar ε-algorithm, of *particular rules* derived by Wynn [39] for this purpose, while the topological ε-algorithms have no particular rules (such rules also exist for other acceleration algorithms [28]). These particular rules are used as soon as, for some fixed p,

$$|\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}|/|\varepsilon_k^{(n)}| < 10^{-p}.$$

When this condition is satisfied, a so-called *singularity* occurs, the particular rule is used instead of the normal one for computing the term of the table possibly affected by numerical instability, and a counter σ indicating the number of singularities detected is increased by 1.

Let us give two numerical examples which show the gain in numerical stability and accuracy brought by the simplified topological ε-algorithms with the particular rules of the scalar ε-algorithm. We denote by STEA1–1, STEA1–2, STEA1–3, and STEA1–4, respectively, the four rules (5.1)–(5.4) of the first simplified topological ε-algorithm, and we use similar notation for the equivalent forms of the second simplified topological ε-algorithm given in section 5.2.

We consider the sequence of vectors defined by

$$\mathbf{S}_0 = \mathbf{r},\ \mathbf{S}_1 = (1,\dots,1)^T, \quad \mathbf{S}_2 = (1,\dots,1)^T + 10^{-11}\mathbf{r}, \quad \mathbf{S}_3 = \mathbf{S}_0, \quad \mathbf{S}_4 = \mathbf{S}_0 + 10^{-11}\mathbf{r},$$
$$\mathbf{S}_n = 3\mathbf{S}_{n-1} - \mathbf{S}_{n-2} + 2\mathbf{S}_{n-3} + \mathbf{S}_{n-4} - 5\mathbf{S}_{n-5}, \quad n = 5, 6, \dots,$$

TABLE 2
Performances of the various algorithms for a sequence of vectors.

	$p=12$		$p=13$	
Algorithm	σ	$\\|\varepsilon_{10}^{(0)}\\|_\infty$	σ	$\\|\varepsilon_{10}^{(0)}\\|_\infty$
TEA1		9.92×10^{-1}		9.92×10^{-1}
STEA1–X	2	9.35×10^{-4}	0	1.15
TEA2		6.01		6.01
STEA2–1	2	9.42×10^{-13}	0	3.10
STEA2–2	2	1.67×10^{-12}	0	3.10
STEA2–3	2	9.46×10^{-13}	0	3.10
STEA2–4	2	1.66×10^{-12}	0	3.10

where $\mathbf{r}$ is a fixed random vector whose components are uniformly distributed in $[0,1]$. Since the vectors $\mathbf{S}_n$ satisfy a linear difference equation of order 5 whose constant term is $\mathbf{0}$, we must have, from (3.1) and Theorem 3.2, $\widehat{\varepsilon}_{10}^{(n)}=\widetilde{\varepsilon}_{10}^{(n)}=\mathbf{0}$ for all n. With vectors of dimension 10000 and $\mathbf{y}=(1,\ldots,1)^T$, we obtain the results of Table 2 (STEA1–X denotes any of the forms of the algorithm). The notation $\varepsilon_{10}^{(0)}$ corresponds to $\widehat{\varepsilon}_{10}^{(0)}$ for TEA1 and STEA1–X, and to $\widetilde{\varepsilon}_{10}^{(0)}$ for TEA2 and STEA2–X. We can see the important improvement obtained by STEA2 when $\sigma=2$ singularities have been detected and treated by the particular rules of the scalar ε-algorithm.

TABLE 3
Performances of the various algorithms for a sequence of matrices.

	$p=7$		$p=8$		$p=10$	
Algorithm	σ	$\\|\varepsilon_{10}^{(0)}\\|_\infty$	σ	$\\|\varepsilon_{10}^{(0)}\\|_\infty$	σ	$\\|\varepsilon_{10}^{(0)}\\|_\infty$
STEA1–X	2	6.14×10^{-7}	1	1.04	0	1.04
STEA2–1	2	1.31×10^{-12}	1	3.02	0	3.02
STEA2–2	2	1.05×10^{-12}	1	3.02	0	3.02
STEA2–3	2	1.30×10^{-12}	1	3.02	0	3.02
STEA2–4	2	1.04×10^{-12}	1	3.02	0	3.02

Let us now consider the case where the $\mathbf{S}_n$'s are $m\times m$ matrices constructed exactly by the same recurrence relation as the vectors of the previous example and with the same initializations ($\mathbf{r}$ is now a fixed random matrix). The linear functional $\mathbf{y}$ is defined as $\langle\mathbf{y},\mathbf{S}_n\rangle=\mathrm{trace}(\mathbf{S}_n)$. For $m=2000$, we obtain the results of Table 3. We remark that the treatment of only one singularity is not enough for avoiding numerical instability but that, when $\sigma=2$, both STEA1 and STEA2 give pretty good results, which underlines the importance of detecting the singularities and using the particular rules.

To end this section, we want to add two remarks. In our tests, even if the speed of convergence is the same, the results obtained with STEA2 are often better. This could be due to the fact that STEA2 computes a combination (with the same coefficients) of $\mathbf{S}_{n+k}, \ldots, \mathbf{S}_{n+2k}$ (which are usually closer to $\mathbf{S}$) instead of $\mathbf{S}_n, \ldots, \mathbf{S}_{n+k}$ for STEA1. Also, from the tests performed, we cannot decide which of the four equivalent forms of each simplified algorithm seems to be the most stable one. However, these experiments suggest that we avoid using terms of the scalar ε-algorithm with an odd lower index, and thus, in the next section, where some applications of the topological Shanks transformations are presented, we will show only the results obtained by STEA1–3 and STEA2–3.

8. Applications.

We consider two types of applications, one with sequences of vectors and another one with sequences of matrices.

8.1. System of equations.

There exist many iterative methods for solving systems of nonlinear and linear equations. For nonlinear systems, the ε-algorithms (scalar, vector, topological) lead to methods with a quadratic convergence under some assumptions [9]. For linear systems, these algorithms are, in fact, direct methods, but they cannot be used in practice since the computation of the exact solution requires too much storage. However, they can be used for accelerating the convergence of iterative methods. Let us look at such an example.

Kaczmarz's method [22] is an iterative method for linear equations. It is used, in particular, in tomographic imaging, where it is known as the *Algebraic Reconstruction Technique* (ART). It is well suited for parallel computations and large-scale problems because each step requires only one row of the matrix (or several rows simultaneously in its block version) and no matrix-vector products are needed. It is always converging, but its convergence is often quite slow. Procedures for its acceleration were studied in [10]. For the `parter` matrix (a Cauchy matrix and a Toeplitz matrix with singular values near π and with elements $1/(i-j+0.5)$) of dimension 5000 from the MATLAB `gallery` of test matrices with $\mathbf{x} = (1, \ldots, 1)^T$, $\mathbf{b} = \mathbf{A}\mathbf{x}$ computed accordingly, and $\mathbf{y} = \mathbf{b}$, Kaczmarz's method achieves an error of 3.44×10^{-1} after 48 iterations. With $k = 1, 3$, or 5, all our algorithms produce an error between 10^{-12} and 10^{-13} after 41, 24, and 19 iterations, respectively. The simplified ε-algorithm requires only the computation of $2k+1$ scalar products, some simple arithmetical operations, and the storage of a number of auxiliary vectors, as stated in Table 1. Thanks to the ascending diagonal technique presented in the preceding section, it can be applied simultaneously to the computation of the iterates of Kaczmarz's method. As soon as a new iterate of it is obtained, its scalar product with $\mathbf{y}$ is computed and the iterate is no longer required. Thus, the parallelism of Kaczmarz's method is not affected.

Kaczmarz's method belongs to the class of *Alternating Projection Methods* for finding a point in the intersection of several subsets of an Hilbert space. They are studied in [14], where some procedures for their acceleration are given. The algorithms proposed in this paper could also be helpful in this context.

8.2. Matrix equations.

Matrix equations, such as the algebraic Riccati, Lyapunov, or Sylvester equations, form an important domain of numerical analysis, and they intervene in many applications: Newton's method for computing the inverse of a matrix, or its square root, or the matrix sign function, etc.; see [15] for some of these applications. Let us also mention the computation of matrix functions. The solution of these problems usually requires iterative methods.

To illustrate our algorithms, let us consider some examples where a sequence of square matrices $\mathbf{S}_n \in \mathbb{R}^{m \times m}$ has to be accelerated. In these examples, the duality product with $\mathbf{y}$ corresponds to the trace of the matrix.

Example 1. A new inversion-free iterative method for obtaining the minimal Hermitian positive definite solution of the matrix rational equation $F(\mathbf{S}) = \mathbf{S} + \mathbf{A}^*\mathbf{S}^{-1}\mathbf{A} - \mathbf{I} = \mathbf{0}$, where $\mathbf{I}$ is the identity matrix and $\mathbf{A}$ is a given nonsingular matrix, was proposed in [26]. It consisted of the following iterations, denoted by NS:

$$\mathbf{S}_{n+1} = 2\mathbf{S}_n - \mathbf{S}_n\mathbf{A}^{-*}(\mathbf{I} - \mathbf{S}_n)\mathbf{A}^{-1}\mathbf{S}_n, \quad n = 0, 1, \ldots,$$

with $\mathbf{S}_0 = \mathbf{A}\mathbf{A}^*$. All iterates $\mathbf{S}_n$ are Hermitian. The inverse of $\mathbf{A}$ has to be computed once at the beginning of the iterations, each of them needing three matrix-matrix products since $\mathbf{S}_n\mathbf{A}^{-*}$ is the conjugate transpose of $\mathbf{A}^{-1}\mathbf{S}_n$. However, in practice, rounding errors destroy the Hermitian character of $\mathbf{S}_n$ when n grows. Thus, it is better to program the method as written above without using the conjugate transpose of $\mathbf{A}^{-1}\mathbf{S}_n$, and, thus, each iteration requires four matrix-matrix products, as explained in [27].

We tried the five examples given in [26]. Each of them corresponds to a different nonsingular matrix A of dimension 3 or 4. For example 1, the gain is only one or two iterations. For example 2, the same precision is obtained by STEA1 and STEA2 with $k = 1$ in 19–20 iterations instead of 27 for NS, and in 16 for $k = 2$ and $k = 3$. For examples 3 and 5, no acceleration is obtained since the NS iterations converge pretty well. As proved in [13], a universal algorithm able to accelerate the convergence of all converging sequences cannot exist. Even for large classes of special sequences such an algorithm cannot exist. Thus, a gain can only be guaranteed under theoretical results.

For example 4, we have the results of Table 4. If we denote by m the number of terms of NS used, the results given in this table correspond to the Frobenius norm of $F(\boldsymbol{\varepsilon}_{2k}^{(m-2k)})$ for the simplified ε-algorithms, and to that of $F(\mathbf{S}_m)$ for NS.

TABLE 4
Performances of the various algorithms for the equation $F(\mathbf{S}) = \mathbf{0}$ (Example 1).

	$k=1$		$k=2$		$k=3$	
Algorithm	m	Frob. norm	m	Frob. norm	m	Frob. norm
STEA1	60	1.47×10^{-15}	60	1.52×10^{-15}	43	1.63×10^{-15}
NS	60	6.80×10^{-9}	60	6.80×10^{-9}	43	5.62×10^{-7}
STEA2	61	1.84×10^{-15}	61	1.93×10^{-15}	49	1.61×10^{-15}
NS	61	5.25×10^{-9}	61	5.25×10^{-9}	49	1.18×10^{-7}

Example 2. More generally, consider now the matrix equation $\mathbf{S} + \mathbf{A}^*\mathbf{S}^{-q}\mathbf{A} = \mathbf{Q}$ for $0 < q \leq 1$. It can be solved by the iterative method (2.6) of [43],

$$\begin{aligned} \mathbf{S}_n &= \mathbf{Q} - \mathbf{A}^*\mathbf{S}_n^q\mathbf{A}, \\ \mathbf{Y}_{n+1} &= 2\mathbf{Y}_n - \mathbf{Y}_n\mathbf{S}_n\mathbf{Y}_n, \end{aligned}$$

with $\mathbf{Y}_0 = (\gamma\mathbf{Q})^{-1}$ and γ conveniently chosen. For example 4.2 of dimension 5 with $q = 0.7$ and $\gamma = 0.9985$ treated in [43], the Euclidean norm of the error is 1.30×10^{-10} at iteration 13 of the iterative method, while, with $k = 3$, it is 5.8×10^{-11} for STEA1 and 6.5×10^{-14} for STEA2. At iteration 17, the norm of the error of the iterative method is 4.11×10^{-14}, that of STEA1 is 3.5×10^{-12}, and STEA2 gives 1.49×10^{-14}.

This example illustrates the fact that no improvement is obtained when the iterative method converges pretty well. We also see that STEA2 gives slightly better results than STEA1. However, when full machine precision is reached, the two algorithms are equivalent.

For the same example, but now with $q = 0.5$, we obtain the results of Figure 4(left). The solid line represents the iterative method, the dash-dotted curve is obtained by STEA1, and the dashed one is obtained by STEA2. This example clearly shows the acceleration which can be brought by the algorithms.

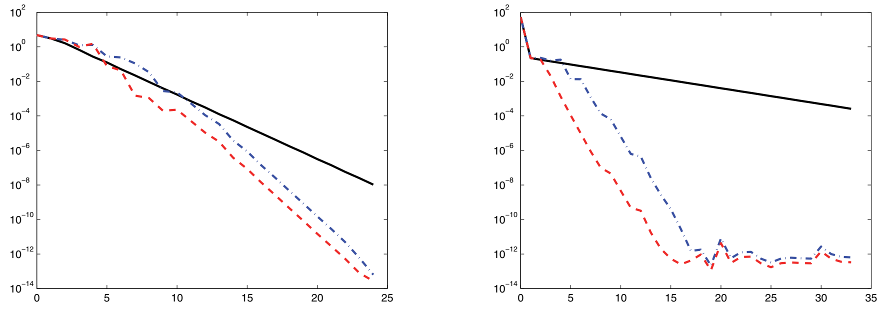

FIG. 4. *Iterative method (solid line),* STEA1 *(dash-dotted line), and* STEA2 *(dashed line) for Examples* 2 *(left) and* 3 *(right).*

Example 3. We consider now the symmetric Stein matrix equation, also called the discrete-time Lyapunov equation, $\mathbf{S} - \mathbf{A}\mathbf{S}\mathbf{A}^T = \mathbf{F}\mathbf{F}^T$, $\mathbf{F} \in \mathbb{R}^{m \times s}$, $s \ll m$, with the eigenvalues of $\mathbf{A}$ inside the unit disk. As explained in [21], this equation can be solved by the iterative method given in [34], $\mathbf{S}_{n+1} = \mathbf{F}\mathbf{F}^T + \mathbf{A}\mathbf{S}_n\mathbf{A}^T$, $n = 0, 1, \ldots,$ with $\mathbf{S}_0 = \mathbf{0}$.

The results of Figure 4(right) correspond to the `moler` matrix for $\mathbf{A}$ divided by an adequate factor so that its spectral radius is equal to 0.9. The `moler` matrix is a symmetric positive definite matrix with one small eigenvalue. Its elements are $a_{ij} = \min(i, j) - 2$ and $a_{ii} = i$. The matrix $\mathbf{F}$ is the `parter` matrix of dimension 500×30. These two matrices are from the MATLAB `gallery` of test matrices. We took $k = 3$ in the simplified algorithms.

9. Conclusions.

To conclude, let us summarize the characteristics of the old and new algorithms for implementing the topological Shanks transformations and compare them.

TOPOLOGICAL ε-ALG. (TEA1, TEA2)	SIMPLIFIED ε-ALG. (STEA1, STEA2)
– two rules;	– only one rule;
– storage of one and a half ascending diagonals for TEA1, and one ascending diagonal for TEA2;	– storage of two half ascending diagonals for STEA1, and half of an ascending diagonal for STEA2;
– storage of elements of E and E^*;	– storage of only elements of E;
– use of the duality product by elements of E^* computed recursively, and by $\mathbf{y}$ inside the rule of the algorithms;	– application of $\mathbf{y}$ only to $\mathbf{S}_n \in E$, and no use of the duality product inside the rules;
– convergence and acceleration results difficult to obtain (rules too complicated);	– possibility of proving convergence and acceleration results;
– numerical instability can be present.	– possibility of improving the numerical stability.

Acknowledgments. We would like to thank Marcos Raydan for exchanges about the iterative methods for solving the matrix rational equations. We also thank the reviewer, whose constructive comments helped us to clarify our explanations and to polish the paper.

REFERENCES

[1] C. Brezinski, *Études sur les ε et ϱ-algorithmes*, Numer. Math., 17 (1971), pp. 153–162.
[2] C. Brezinski, *L'ε algorithme et les suites totalement monotones et oscillantes*, C. R. Acad. Sci. Paris Sér. A-B, 276 (1973), pp. A305–A308.
[3] C. Brezinski, *Généralisation de la transformation de Shanks, de la table de Padé et de l'ε–algorithme*, Calcolo, 12 (1975), pp. 317–360.
[4] C. Brezinski, *Computation of the eigenelements of a matrix by the ε-algorithm*, Linear Algebra and Appl., 11 (1975), pp. 7–20.
[5] C. Brezinski, *Génération de suites totalement monotones et oscillantes*, C. R. Acad. Sci. Paris Sér. A-B, 280 (1975), pp. A729–A731.
[6] C. Brezinski, *Padé-Type Approximation and General Orthogonal Polynomials*, Internat. Ser. Numer. Math. 50, Birkhäuser-Verlag, Basel, 1980.
[7] C. Brezinski, *A general extrapolation algorithm*, Numer. Math., 35 (1980), pp. 175–187.
[8] C. Brezinski and M. Crouzeix, *Remarques sur le procédé Δ^2 d'Aitken*, C. R. Acad. Sci. Paris Sér. A-B, 270 (1970), pp. A896–A898.
[9] C. Brezinski and M. Redivo-Zaglia, *Extrapolation Methods. Theory and Practice*, North–Holland, Amsterdam, 1991.
[10] C. Brezinski and M. Redivo-Zaglia, *Convergence acceleration of Kaczmarz's method*, J. Engrg. Math., DOI 10.1007/s10665-013-9656-3, published online, August 2013.
[11] C. Brezinski and H. Sadok, *Lanczos-type algorithms for solving systems of linear equations*, Appl. Numer. Math., 11 (1993), pp. 443–473.
[12] C. Brezinski and G. Walz, *Sequences of transformations and triangular recursion schemes, with applications in numerical analysis*, J. Comput. Appl. Math., 34 (1991), pp. 361–383.
[13] J. P. Delahaye and B. Germain-Bonne, *Résultats négatifs en accélération de la convergence*, Numer. Math., 35 (1980), pp. 443–457.
[14] R. Escalante and M. Raydan, *Alternating Projection Methods*, SIAM, Philadelphia, 2011.
[15] A. Frommer and V. Simoncini, *Matrix functions*, in Model Order Reduction: Theory, Research Aspects and Applications, Math. Ind. 13, W. H. Schilders, H. A. van der Vorst, and J. Rommes, eds., Springer, Heidelberg, 2008, pp. 275–304.
[16] C. R. Garibotti and F. F. Grinstein, *Recent results relevant to the evaluation of infinite series*, J. Comput. Appl. Math., 9 (1983), pp. 193–200.
[17] P. R. Graves-Morris and C. D. Jenkins, *Vector-valued rational interpolants* III, Constr. Approx., 2 (1986), pp. 263–289.
[18] K. Jbilou, A. Messaoudi, and K. Tabaa, *On some matrix extrapolation methods*, C. R. Math. Acad. Sci. Paris, 341 (2005), pp. 781–786.
[19] K. Jbilou, L. Reichel, and H. Sadok, *Vector extrapolation enhanced TSVD for linear discrete ill-posed problems*, Numer. Algorithms, 51 (2009), pp. 195–208.
[20] K. Jbilou and H. Sadok, *Vector extrapolation methods. Applications and numerical comparison*, J. Comput. Appl. Math., 122 (2000), pp. 149–165.
[21] K. Jbilou and H. Sadok, *Matrix polynomial and epsilon-type extrapolation methods with applications*, Numer. Algorithms, DOI 10.1007/s11075-014-9879-z, published online, June 2014.
[22] S. Kaczmarz, *Angenäherte Auflösung von Systemen linearer Gleichungen*, Bull. Acad. Polon. Sci., A35 (1937), pp. 355–357 (in German); Int. J. Control, 57 (1993), pp. 1269–1271 (in English).
[23] A. C. Matos, *Acceleration results for the vector E-algorithm*, Numer. Algorithms, 1 (1991), pp. 237–260.
[24] J. B. McLeod, *A note on the ε-algorithm*, Computing, 7 (1971), pp. 17–24.
[25] P. Midy, *Scaling transformations and extrapolation algorithms for vector sequences*, Comput. Phys. Comm., 70 (1992), pp. 285–291.
[26] M. Monsalve and M. Raydan, *A new inversion-free method for a rational matrix equation*, Linear Algebra Appl., 433 (2010), pp. 64–71.
[27] M. Monsalve and M. Raydan, *Corrigendum to "A new inversion-free method for a rational matrix equation,"* Linear Algebra Appl., 448 (2014), pp. 343–344.

[28] M. Redivo-Zaglia, *Particular rules for the Θ-algorithm*, Numer. Algorithms, 3 (1992), pp. 353–369.

[29] A. Salam, *Non-commutative extrapolation algorithms*, Numer. Algorithms, 7 (1994), pp. 225–251.

[30] A. Salam and P. R. Graves-Morris, *On the vector ε-algorithm for solving linear systems of equations*, Numer. Algorithms, 29 (2002), pp. 229–247.

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

[32] D. A. Smith, W. F. Ford, and A. Sidi, *Extrapolation methods for vector sequences*, SIAM Rev., 29 (1987), pp. 199–233.

[33] D. A. Smith, W. F. Ford, and A. Sidi, *Erratum: Correction to "Extrapolation methods for vector sequences,"* SIAM Rev., 30 (1988), pp. 623–624.

[34] R. A. Smith, *Matrix equation $XA + BX = C$*, SIAM J. Appl. Math., 16 (1968), pp. 198–201.

[35] R. C. E. Tan, *Implementation of the topological ε-algorithm*, SIAM J. Sci. Statist. Comput., 9 (1988), pp. 839–848.

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

[37] P. Wynn, *Acceleration techniques for iterated vector and matrix problems*, Math. Comp., 16 (1962), pp. 301–322.

[38] P. Wynn, *Acceleration techniques in numerical analysis, with particular reference to problems in one independent variable*, in Proceedings of IFIP Congress 62, North–Holland, Amsterdam, 1962, pp. 149–156.

[39] P. Wynn, *Singular rules for certain nonlinear algorithms*, BIT, 3 (1963), pp. 175–195.

[40] P. Wynn, *An arsenal of Algol procedures for the evaluation of continued fractions and for effecting the epsilon algorithm*, Chiffres, 4 (1966), pp. 327–362.

[41] P. Wynn, *On the convergence and stability of the epsilon algorithm*, SIAM J. Numer. Anal., 3 (1966), pp. 91–122.

[42] P. Wynn, *Sur les suites totalement monotones*, C. R. Acad. Sci. Paris Sér. A-B, 275 (1972), pp. A1065–A1068.

[43] X. Yin, S. Liu, and T. Li, *On positive definite solutions of the matrix equation $X + A^*X^{-q}A = Q (0 < q \leq 1)$*, Taiwanese J. Math., 16 (2012), pp. 1391–1407.