ELSEVIER

Journal of Computational and Applied Mathematics 98 (1998) 149–175

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICS

Vector sequence transformations: Methodology and applications to linear systems

C. Brezinski *

Laboratoire d'Analyse Numérique et d'Optimisation, Université des Sciences et Technologies de Lille, 59655-Villeneuve d'Ascq cedex, France

Received 15 April 1998

Abstract

In this paper, a methodology for the construction of various vector sequence transformations is formulated leading to a unified presentation of the subject and to new results. The connections to the general interpolation problem and to projections are discussed. Various particular cases are examined in more details. Applications to the solution of systems of linear equations will end the paper and, in particular, their relation with Lanczos method will be studied. Some numerical examples will be given. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Vector sequence transformations; Convergence acceleration; Systems of linear equations; Lanczos method

0. Introduction

Let (x_n) be a sequence of vectors of $\mathbb{R}^p$ or $\mathbb{C}^p$ (or, more generally, a sequence of elements of a real or complex vector space) assumed to converge to x. If this sequence converges too slowly, then it will be transformed, by a vector sequence transformation T, into another sequence of vectors, say (y_n), in order to try to accelerate its convergence. There exist many vector sequence transformations (see [16] for a review): the vector ε-algorithm [55], Henrici's transformation [37, p.116], the topological ε-algorithm [3], the MMPE [3, 41], the MPE [25], Germain–Bonne's transformation [33], the RRE [28], the vector E-algorithm [6], the RPA and its variants [9], the vector Padé approximants [51, 52], the H-algorithm [20], the vector G-transform [16, p.243], the $S\beta$-algorithm [39], the VTT and the BVTT [18], the multiparameter Richardson acceleration method [15], and others.

The aim of this paper is to give a general methodology for the construction of vector sequence transformations. This synthetic presentation makes clearer the similarities and the differences between them. Moreover, some new or wider results will be obtained and a more profound understanding

* E-mail: Claude.Brezinski@univ-lille1.fr.

PII: S0377-0427(98)00119-8

will be gained. The connections between these transformations, the general interpolation problem and projections will be studied. Applications to the construction and the acceleration of iterative methods for solving systems of linear equations will be discussed in the last section. In particular, relations with Lanczos method will be emphasized. Some numerical examples will be given to illustrate one of the acceleration methods.

Usually, when constructing a method for accelerating the convergence of sequences, the starting point is the *kernel* of the transformation, that is the set of sequences which are transformed into a constant sequence. The kernel depends of some parameters which are determined by solving a system of linear equations. It follows that the terms y_n of the new sequence can be written as ratios of two determinants or, equivalently, as Schur complements. Then, the expressions of some auxiliary matrices appearing in these Schur complements have to be found. Next, recursive algorithms for computing the y_n's without computing the determinants have to be obtained. Finally, from the ratios of determinants, these transformations could be connected to the general interpolation problem and to projection.

In this paper, we will follow this procedure for a quite general type of kernel. Other types of kernels are studied in [23].

1. Construction of the transformations

We want to construct a vector sequence transformation $T:(x_n)\mapsto(y_n)$ such that $\forall n, y_n=x$ if (and, sometimes, only if) $(x_n)\in N_T^k$, a set of sequences called the *kernel* of T. In this paper, we will consider kernels of the form

$$N_T^k=\{(x_n)\,|\,\forall n,\ x_n=x+a_1z_n^1+\cdots+a_kz_n^k\},$$

where x is an unknown vector, the a_i's unknown scalars and the (z_n^i) known sequences of vectors. Sometimes, in the sequel, y_n will be denoted by y_n^k when necessary. It is obvious that, if $N_T^{k+1}=N_T^k+\operatorname{span}(z_n^{k+1})$, then $N_T^k\subset N_T^{k+1}$. Let us denote by Z_n the $p\times k$ matrix whose columns are $z_n^1,\dots,z_n^k$ and set $\alpha=(a_1,\dots,a_k)^{\mathrm{T}}$. Then, a sequence (x_n) in the kernel N_T^k can be written as

$$x_n=x+Z_n\alpha,\quad n=0,1,\dots$$

Let us first explain the methodology for constructing a vector sequence transformation with this kernel. If the scalars $a_1,\dots,a_k$ were known, then, by construction, the transformation T defined by

$$y_n=x_n-a_1z_n^1-\cdots-a_kz_n^k,\quad n=0,1,\dots \tag{1}$$

will have the kernel N_T^k. Thus, the problem is to compute the scalars a_i. We can first eliminate x by writing

$$\Delta x_n=a_1\Delta z_n^1+\cdots+a_k\Delta z_n^k, \tag{2}$$

where the forward difference operator Δ acts on the lower index n.

For obtaining a system of k linear equations in the k unknowns $a_1,\dots,a_k$, several strategies can be used

1. the E strategy: we write down (2) for the indexes $n,\dots,n+k-1$ and multiply scalarly each of them by a vector u;
2. the Θ strategy: we write down (2) for the index n and multiply it scalarly by k linearly independent vectors $u_1,\dots,u_k$;
3. the mixed strategies: we write down (2) for some indexes $n, n+1,\dots$ and multiply them scalarly by linearly independent vectors $u_1, u_2,\dots$ in order to get, in total, k scalar equations. As we will see, there are two main such strategies.

Remark 1. Let us mention that, in the following theoretical results about the various strategies, the vector u and the vectors $u_1,\dots,u_k$ can depend on n. However, in the algorithms for the recursive computation of the vectors y_n for increasing values of k, they are assumed to be fixed vectors.

The E strategy was introduced in [6] for constructing the vector E-algorithm. A particular case of the Θ strategy was mentioned in [3] but it was only fully developed in [18]. It is a generalization of the Θ-procedure, a special device for constructing sequence transformations [8]. One of the mixed strategies is related to vector Padé approximants [51] while the second one seems to be new.

If the sequence (x_n) to be transformed does not belong to N_T^k, we will look for $(x'_n = x' + a_1 z_n^1 + \cdots + a_k z_n^k) \in N_T^k$ such that $x'_n = x_n$ for the same indices n as those used in the strategy and, then compute x' as described above and, finally, approximate the limit x of (x_n) by $y_n = x'$. If, for $i=1,\dots,k$, the sequences (z_n^i) converge to zero, then x' is the limit of the sequence (x'_n). Thus, a sequence transformation is an extrapolation method. It must be understood that, although the kernels of the vector sequence transformations obtained by all the preceding strategies are the same, these transformations do not give identical results when applied to a sequence not belonging to their common kernel and that their acceleration properties are different.

Let us now recall the notion of Schur complement and its relation to ratios of determinants. We consider the partitioned square matrix

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

where D is a square submatrix assumed to be nonsingular. The Schur complement of D in M, denoted by (M/D), is the matrix

$$(M/D) = A - BD^{-1}C$$

and we have

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I & BD^{-1} \\ 0 & I \end{pmatrix} \begin{pmatrix} (M/D) & 0 \\ C & D \end{pmatrix}.$$

It follows

$$\det(M/D) = \frac{\det M}{\det D}.$$

The notion of Schur complement and the preceding determinantal identity are still valid in the case where $A \in \mathbb{R}^p$, $B \in \mathbb{R}^{p\times k}$, $C \in \mathbb{R}^k$ and $D \in \mathbb{R}^{k\times k}$. In this situation, $\det M$ is the vector of $\mathbb{R}^p$ obtained by expanding M with respect to its first vector row (formed by the vector A and the k

vectors which are the columns of the matrix B) by the classical rules for expanding a determinant. For example, if v_1, v_2 and v_3 are vectors and a,b,c,d,e and f scalars, it means

$$\begin{vmatrix} v_1 & v_2 \\ a & b \end{vmatrix} = bv_1 - av_2, \qquad \begin{vmatrix} v_1 & v_2 & v_3 \\ a & b & c \\ d & e & f \end{vmatrix} = v_1 \begin{vmatrix} b & c \\ e & f \end{vmatrix} - v_2 \begin{vmatrix} a & c \\ d & f \end{vmatrix} + v_3 \begin{vmatrix} a & b \\ d & e \end{vmatrix}.$$

Let us assume that $A = x \in \mathbb{R}$. Then $B = (z_1, \ldots, z_k)$ is a row of numbers and (M/D) is also a number. Thus, in this case, $(M/D) = \det(M/D) = \det M / \det D$. Now if $A = x \in \mathbb{R}^p$ and if $z_i \in \mathbb{R}^p$, we have a similar result since the preceding equality holds for each component of the vectors $M, (M/D), \det(M/D) \in \operatorname{span}(x, z_1, \ldots, z_k)$, as explained above. Thus we proved the following result which is fundamental for the sequel

Lemma 1.1. *If $A = x \in \mathbb{R}^p$ and B is the vector row $(z_1, \ldots, z_k)$ with $z_i \in \mathbb{R}^p$, then*

$$(M/D) = \frac{\det M}{\det D}.$$

Obviously the case of complex vectors can be treated similarly.

1.1. The E strategy

Let u be a given vector. Multiplying scalarly (2) by u and writing it down for the indexes $n, \ldots, n+k-1$, we obtain $a_1, \ldots, a_k$ as the solution of the linear system (assumed to be nonsingular)

$$(u, \Delta x_{n+j}) = a_1(u, \Delta z^1_{n+j}) + \cdots + a_k(u, \Delta z^k_{n+j}), \quad j = 0, \ldots, k-1. \tag{3}$$

Solving this system and using (1), we get

$$y_n = \begin{vmatrix} x_n & z_n^1 & \cdots & z_n^k \\ (u, \Delta x_n) & (u, \Delta z_n^1) & \cdots & (u, \Delta z_n^k) \\ \vdots & \vdots & & \vdots \\ (u, \Delta x_{n+k-1}) & (u, \Delta z^1_{n+k-1}) & \cdots & (u, \Delta z^k_{n+k-1}) \end{vmatrix} \Bigg/ \begin{vmatrix} (u, \Delta z_n^1) & \cdots & (u, \Delta z_n^k) \\ \vdots & & \vdots \\ (u, \Delta z^1_{n+k-1}) & \cdots & (u, \Delta z^k_{n+k-1}) \end{vmatrix}. \tag{4}$$

By construction, $\forall n, y_n = x$ if $(x_n) \in N_T^k$. In fact, under some assumptions, this condition is also necessary as we will see below.

Let us denote by X_n the vector with components $(u, x_n), \ldots, (u, x_{n+k-1})$, and by D_n the matrix whose determinant is the denominator of y_n. By the determinantal identity for the Schur complement,

$$y_n = x_n - Z_n D_n^{-1} \Delta X_n.$$

holds.

Let us now give the expressions of the matrix D_n and of the vector ΔX_n. We set

$$Z_n^k = \begin{pmatrix} Z_n \\ \vdots \\ Z_{n+k-1} \end{pmatrix}, \quad U_k = \begin{pmatrix} u & & \\ & \ddots & \\ & & u \end{pmatrix}, \quad X_n^k = \begin{pmatrix} x_n \\ \vdots \\ x_{n+k-1} \end{pmatrix}.$$

Z_n^k and U_k are $kp \times k$ matrices and $X_n^k \in \mathbb{R}^{kp}$. We have $D_n = U_k^T \Delta Z_n^k$ and $\Delta X_n = U_k^T \Delta X_n^k$ and it follows

$$y_n = x_n - Z_n (U_k^T \Delta Z_n^k)^{-1} U_k^T \Delta X_n^k = \frac{\begin{vmatrix} x_n & Z_n \\ U_k^T \Delta X_n^k & U_k^T \Delta Z_n^k \end{vmatrix}}{|U_k^T \Delta Z_n^k|}. \tag{5}$$

Let us now prove the main result about the kernel of this transformation.

Theorem 1.2. *If $(x_n) \in N_T^k$ and if $\forall n$, the matrix $U_k^T \Delta Z_n^k$ is nonsingular, then, $\forall n, y_n = x$. If $\forall n, y_n = x$ and if, $\forall n$, the matrix $U_k^T Z_n^k$ is nonsingular, then $(x_n) \in N_T^k$.*

Proof. The first result follows from the construction of the transformation and from (5).

Let us prove the second part. We remark that y_n exists only if $U_k^T \Delta Z_n^k$ is nonsingular. We have $y_n = x_n - Z_n \Lambda_n$ with $\Lambda_n = (U_k^T \Delta Z_n^k)^{-1} U_k^T \Delta X_n^k$. Thus, if $\forall n, y_n = x$, then $\Delta x_n = \Delta(Z_n \Lambda_n)$. From the expression of Λ_n and from the definition of X_n^k, it follows

$$(U_k^T \Delta Z_n^k) \Lambda_n = U_k^T \Delta X_n^k = U_k^T \Delta(Z_n^k \Lambda_n).$$

Thus $U_k^T Z_{n+1}^k \Lambda_n = U_k^T Z_{n+1}^k \Lambda_{n+1}$ and, then, $\Lambda_n = \Lambda_{n+1}$ if the matrix $U_k^T Z_{n+1}^k$ is nonsingular. □

Remark 1.3. Theorem 1.2 also holds for the transformations defined by $y_n = x_{n+i} - Z_{n+i} D_n^{-1} \Delta X_n$ where $i = 0, \ldots, k$.

When k increases, the vectors y_n (that will now be denoted by y_n^k) can be recursively computed by the vector E-algorithm [6] in the case where the vector u does not depend on n,

$$\begin{aligned} y_n^k &= y_n^{k-1} - \frac{(u, \Delta y_n^{k-1})}{(u, \Delta z_n^{k-1,k})} z_n^{k-1,k}, \\ z_n^{k,i} &= z_n^{k-1,i} - \frac{(u, \Delta z_n^{k-1,i})}{(u, \Delta z_n^{k-1,k})} z_n^{k-1,k}, \quad i > k \end{aligned} \tag{6}$$

with $y_n^0 = x_n$ and $z_n^{0,i} = z_n^i$ and where the operator Δ acts on the lower indexes n.

Remark 1.4. As shown by Wimp [54, p.176], (4) can also be written as

$$y_n^k = \begin{vmatrix} x_n & 0 & z_n^1 & \cdots & z_n^k \\ (u, x_n) & 1 & (u, z_n^1) & \cdots & (u, z_n^k) \\ \vdots & \vdots & \vdots & & \vdots \\ (u, x_{n+k}) & 1 & (u, z_{n+k}^1) & \cdots & (u, z_{n+k}^k) \end{vmatrix} \Bigg/ \begin{vmatrix} 1 & (u, z_n^1) & \cdots & (u, z_n^k) \\ \vdots & \vdots & & \vdots \\ 1 & (u, z_{n+k}^1) & \cdots & (u, z_{n+k}^k) \end{vmatrix}.$$

If u is replaced by u_n in the second row of the numerator and in the first row of the denominator, by u_{n+1} in the third row of the numerator and in the second row of the denominator, and so on, a recursive algorithm for the computation of the vectors y_n^k was obtained by Wimp [54, p.177]. It is a generalization of the vector E-algorithm.

1.2. The Θ strategy

Let $u_1,\dots,u_k$ be k given linearly independent vectors (which is only possible if $k \leqslant p$, the dimension of the vectors). Multiplying scalarly (2) by $u_1,\dots,u_k$, we obtain $a_1,\dots,a_k$ as the solution of the linear system (assumed to be nonsingular)

$$(u_i,\Delta x_n)=a_1(u_i,\Delta z_n^1)+\cdots+a_k(u_i,\Delta z_n^k), \quad i=1,\dots,k. \tag{7}$$

Solving this system and using (1), we get

$$y_n=\begin{vmatrix} x_n & z_n^1 & \cdots & z_n^k \\ (u_1,\Delta x_n) & (u_1,\Delta z_n^1) & \cdots & (u_1,\Delta z_n^k) \\ \vdots & \vdots & & \vdots \\ (u_k,\Delta x_n) & (u_k,\Delta z_n^1) & \cdots & (u_k,\Delta z_n^k) \end{vmatrix} \Bigg/ \begin{vmatrix} (u_1,\Delta z_n^1) & \cdots & (u_1,\Delta z_n^k) \\ \vdots & & \vdots \\ (u_k,\Delta z_n^1) & \cdots & (u_k,\Delta z_n^k) \end{vmatrix}. \tag{8}$$

By construction, $\forall n, y_n=x$ if $(x_n)\in N_T^k$.

Let us denote by X_n the vector with components $(u_1,x_n),\dots,(u_k,x_n)$, and by D_n the matrix whose determinant is the denominator of y_n. Then, by the determinantal identity for the Schur complement, we have

$$y_n=x_n-Z_n\,D_n^{-1}\,\Delta X_n.$$

Let U_k be the matrix with columns $u_1,\dots,u_k$. Then $\Delta X_n=U_k^{\mathrm{T}}\Delta x_n$ and $D_n=U_k^{\mathrm{T}}\Delta Z_n$. It follows

$$y_n=x_n-Z_n(U_k^{\mathrm{T}}\Delta Z_n)^{-1}U_k^{\mathrm{T}}\Delta x_n=\frac{\begin{vmatrix} x_n & Z_n \\ U_k^{\mathrm{T}}\Delta x_n & U_k^{\mathrm{T}}\Delta Z_n \end{vmatrix}}{|U_k^{\mathrm{T}}\Delta Z_n|}.$$

This transformation was introduced in [18] and it was called the VTT (vector theta-type transform). For this transformation, Theorem 1.2 is still valid as proved in [18].

Let us give a recursive algorithm for the implementation of the VTT when k increases and when the matrix Z_n corresponding to $k+1$ (now denoted by Z_n^{k+1}) is obtained from Z_n^k by adding the new $(k+1)$th column z_n^{k+1}. Let a^k be the solution of the $k\times k$ system (7) and a^{k+1} the solution of the same system when the dimension is $k+1$. Thus, the system of dimension $k+1$ is obtained by bordering the matrix of the system of dimension k by a new row and a new column and by adding a new component to its right-hand side. Thus, the bordering method [27] (see also [29, pp. 163–168; 30, pp. 105–111; 14, p. 71]) can be used for its solution and we get (the operator Δ always acts on the lower index n)

$$y_n^{k+1}=y_n^k-\beta_n^k\,t_n^k,$$

with

$$\begin{aligned} t_n^k &= z_n^{k+1}-Z_n^k\,b^k,\\ b^k &= (U_k^{\mathrm{T}}\Delta Z_n^k)^{-1}U_k^{\mathrm{T}}\Delta z_n^{k+1},\\ \beta_n^k &= (u_{k+1},\Delta x_n-\Delta Z_n^k\,a^k)/\beta^k,\\ \beta^k &= (u_{k+1},\Delta z_n^{k+1}-\Delta Z_n^k\,b^k). \end{aligned}$$

We see that t_n^k is the vector obtained by applying the VTT to z_n^{k+1} with the same u_i and the same z_n^i as for computing y_n^k.

The bordering method allows to compute recursively the vectors b^k since

$$(U_{k+1}^{\mathrm{T}}\Delta Z_n^{k+1})^{-1} = \begin{pmatrix} (I + b^k u_{k+1}^{\mathrm{T}}\Delta Z_n^k)(U_k^{\mathrm{T}}\Delta Z_n^k)^{-1}/\beta^k & -b^k/\beta^k \\ -u_{k+1}^{\mathrm{T}}\Delta Z_n^k(U_k^{\mathrm{T}}\Delta Z_n^k)^{-1}/\beta^k & 1/\beta^k \end{pmatrix}.$$

We have

$$a^k = (U_k^{\mathrm{T}}\Delta Z_n^k)^{-1}U_k^{\mathrm{T}}\Delta x_n$$

and the preceding formula also allows to compute recursively the vectors a^k by

$$a^{k+1} = \begin{pmatrix} a^k \\ 0 \end{pmatrix} + \beta_n^k \begin{pmatrix} -b^k \\ 1 \end{pmatrix}$$

and, then, we have

$$y_n^{k+1} = x_n - Z_n^{k+1}a^{k+1}.$$

A similar formula for b^{k+1} does not hold since the right-hand side of the linear system giving it is not obtained by adding a new component to the previous one.

A FORTRAN subroutine for the bordering method can be found in [16]. For vector sequence transformations, k is usually small and, so, the storage and the computation of the matrices $(U_k^{\mathrm{T}}\Delta Z_n^k)^{-1}$ is not an obstacle.

Let us also mention that the VTT can be implemented by a recursive algorithm due to Ford and Sidi [31] as explained in [18].

If the matrix U_k (which is now assumed to depend on n) is such that $\forall n, U_k^{\mathrm{T}}Z_{n+1} = 0$, the corresponding transformation is called the BVTT where B stands for *biorthogonal*. The main interest of this transformation compared to the VTT is the following result proved in [18]

Theorem 1.5. *If $k < p$ and if $\forall n$, the matrix $U_k^{\mathrm{T}}Z_n$ is nonsingular and $U_k^{\mathrm{T}}Z_{n+1} = 0$, then the kernel of the BVTT is the set of sequences of the form $x_n = x + Z_n\alpha_n$ where $\alpha_n = (a_1(n), \ldots, a_k(n))^{\mathrm{T}}$.*

This is the only known transformation with a kernel whose unknown constants a_i have been replaced by sequences $(a_i(n))$ depending on n.

For the BVTT, we have

$$\beta_n^k = \frac{(u_{k+1}, y_n^k - x_{n+1})}{(u_{k+1}, t_n^k)}.$$

The preceding recursive algorithm for computing the vectors y_n^k simplifies and, as proved in [18], it is connected to the RIA [9] (see Section 4).

As seen above, the kernel N_T^k is the set of sequences of the form $x_n = x + Z_n\alpha$ where $\alpha = (a_1, \ldots, a_k)^{\mathrm{T}}$. Thus $\Delta x_n = \Delta Z_n\alpha$. Solving this system in the least-squares sense leads to

$$\alpha = [(\Delta Z_n)^{\mathrm{T}}\Delta Z_n]^{-1}(\Delta Z_n)^{\mathrm{T}}\Delta x_n$$

and we get

$$y_n = x_n - Z_n[(\Delta Z_n)^{\mathrm{T}} \Delta Z_n]^{-1} (\Delta Z_n)^{\mathrm{T}} \Delta x_n$$

which corresponds to the VTT with $U_k = \Delta Z_n$.

1.3. The mixed strategies

In a mixed strategy, the relation (2) is multiplied scalarly by u_i and then used for $i = 1, 2, \ldots$ and for $n, n+1, \ldots$ in order to obtain k equations in total. We can write it as

$$(u_i, \Delta x_{n+j}) = a_1(u_i, \Delta z^1_{n+j}) + \cdots + a_k(u_i, \Delta z^k_{n+j}). \tag{9}$$

Thus, we have a great flexibility in our choices of the indexes i and j. However, since the next problem will be to obtain recurrence relationships for computing the vectors y^k_n for increasing values of k, two strategies seem to be more appropriate.

The first strategy has been called $E\Theta$ since the choice of the relations (9) to be used mainly depends on the index i (as in the E strategy) while it is the reverse for the second strategy, called ΘE.

Let $d \geqslant 1$ be a *fixed* integer. We define m as the integer part of k/d and r as the remainder, that is $k = md + r$ with $0 \leqslant r < d$.

For some value of d and k, the two startegies coincide.

1.3.1. The $E\Theta$ strategy

In this strategy, we will make use of (9) for $i = 1, \ldots, d$ and $j = 0, \ldots, m-1$, and also for $i = 1, \ldots, r$ and $j = m$. This strategy is the same as the strategy used for defining the vector Padé approximants [51].

Solving the system (9) for these indexes and using (1), we get

$$y_n = \begin{vmatrix} x_n & z^1_n & \cdots & z^k_n \\ (u_1, \Delta x_n) & (u_1, \Delta z^1_n) & \cdots & (u_1, \Delta z^k_n) \\ \vdots & \vdots & & \vdots \\ (u_d, \Delta x_n) & (u_d, \Delta z^1_n) & \cdots & (u_d, \Delta z^k_n) \\ \vdots & \vdots & & \vdots \\ (u_1, \Delta x_{n+m-1}) & (u_1, \Delta z^1_{n+m-1}) & \cdots & (u_1, \Delta z^k_{n+m-1}) \\ \vdots & \vdots & & \vdots \\ (u_d, \Delta x_{n+m-1}) & (u_d, \Delta z^1_{n+m-1}) & \cdots & (u_d, \Delta z^k_{n+m-1}) \\ (u_1, \Delta x_{n+m}) & (u_1, \Delta z^1_{n+m}) & \cdots & (u_1, \Delta z^k_{n+m}) \\ \vdots & \vdots & & \vdots \\ (u_r, \Delta x_{n+m}) & (u_r, \Delta z^1_{n+m}) & \cdots & (u_r, \Delta z^k_{n+m}) \end{vmatrix} \Bigg/ \det D_n,$$

where D_n is the matrix obtained by deleting the first row and the first column of the matrix whose determinant forms the numerator.

By construction, $\forall n$, $y_n = x$ if $(x_n) \in N^k_T$.

If $d=1$, the E strategy and the corresponding determinantal expression for y_n are recovered. Let us now give the expression of y_n which uses the Schur complement. We set

$$Z_n^k = \begin{pmatrix} Z_n \\ \vdots \\ Z_n \\ \vdots \\ Z_{n+m-1} \\ \vdots \\ Z_{n+m-1} \\ Z_{n+m} \\ \vdots \\ Z_{n+m} \end{pmatrix}, \qquad U_k = \begin{pmatrix} u_1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & u_d & & & & & & \\ & & & \ddots & & & & & \\ & & & & u_1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & u_d & & \\ & & & & & & & u_1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & u_r \end{pmatrix}.$$

In Z_n^k, each Z_{n+i} is repeated d times for $i=0,\ldots,m-1$, and r times for $i=m$. The vector X_n^k has the same structure as Z_n^k, the Z_{n+i} being replaced by the x_{n+i}. On the diagonal of U_k, $u_1,\ldots,u_d$ is repeated m times. So, U_k and Z_n^k are $kp \times k$ matrices and $X_n^k \in \mathbb{R}^{kp}$.

With these notations, we have

$$y_n = x_n - Z_n(U_k^{\mathrm{T}} \Delta Z_n^k)^{-1} U_k^{\mathrm{T}} \Delta X_n^k = \frac{\begin{vmatrix} x_n & Z_n \\ U_k^{\mathrm{T}} \Delta X_n^k & U_k^{\mathrm{T}} \Delta Z_n^k \end{vmatrix}}{|U_k^{\mathrm{T}} \Delta Z_n^k|}.$$

From this expression, we see that Theorem 1.2 also holds for this transformation.

1.3.2. The ΘE strategy

We will now use (9) for $i=1,\ldots,m$ and $j=0,\ldots,d-1$, and also for $i=m+1$ and $j=0,\ldots,r-1$. This strategy seems to be new.

Solving the system (9) for these indexes and using (1), we get

$$y_n = \begin{vmatrix} x_n & z_n^1 & \cdots & z_n^k \\ (u_1, \Delta x_n) & (u_1, \Delta z_n^1) & \cdots & (u_1, \Delta z_n^k) \\ \vdots & \vdots & & \vdots \\ (u_1, \Delta x_{n+d-1}) & (u_1, \Delta z_{n+d-1}^1) & \cdots & (u_1, \Delta z_{n+d-1}^k) \\ \vdots & \vdots & & \vdots \\ (u_m, \Delta x_n) & (u_m, \Delta z_n^1) & \cdots & (u_m, \Delta z_n^k) \\ \vdots & \vdots & & \vdots \\ (u_m, \Delta x_{n+d-1}) & (u_m, \Delta z_{n+d-1}^1) & \cdots & (u_m, \Delta z_{n+d-1}^k) \\ (u_{m+1}, \Delta x_n) & (u_{m+1}, \Delta z_n^1) & \cdots & (u_{m+1}, \Delta z_n^k) \\ \vdots & \vdots & & \vdots \\ (u_{m+1}, \Delta x_{n+r-1}) & (u_{m+1}, \Delta z_{n+r-1}^1) & \cdots & (u_{m+1}, \Delta z_{n+r-1}^k) \end{vmatrix} \Bigg/ \det D_n,$$

where D_n is the matrix obtained by deleting the first row and the first column of the matrix whose determinant forms the numerator.

By construction, $\forall n, y_n = x$ if $(x_n) \in N_T^k$.

If $d = 1$, the Θ strategy and the corresponding determinantal expression for y_n are recovered.

Let us now give the expression of y_n which uses the Schur complement. We set

$$Z_n^k = \begin{pmatrix} Z_n \\ \vdots \\ Z_{n+d-1} \\ \vdots \\ Z_n \\ \vdots \\ Z_{n+d-1} \\ Z_n \\ \vdots \\ Z_{n+r-1} \end{pmatrix}, \qquad U_k = \begin{pmatrix} u_1 & & & & & & & \\ & \ddots & & & & & & \\ & & u_1 & & & & & \\ & & & \ddots & & & & \\ & & & & u_m & & & \\ & & & & & \ddots & & \\ & & & & & & u_m & & & \\ & & & & & & & u_{m+1} & \\ & & & & & & & & \ddots & \\ & & & & & & & & & u_{m+1} \end{pmatrix}.$$

In Z_n^k, each block $Z_n, \ldots, Z_{n+d-1}$ is repeated m times. The vector X_n^k has the same structure as Z_n^k, the Z_{n+i} being replaced by the x_{n+i}. On the diagonal of U_k, each u_i is repeated d times for $i = 1, \ldots, m$, and r times for $i = m+1$. So, U_k and Z_n^k are $kp \times k$ matrices and $X_n^k \in \mathbb{R}^{kp}$.

With these notations, we again have

$$y_n = x_n - Z_n (U_k^{\mathrm{T}} \Delta Z_n^k)^{-1} U_k^{\mathrm{T}} \Delta X_n^k = \frac{\begin{vmatrix} x_n & Z_n \\ U_k^{\mathrm{T}} \Delta X_n^k & U_k^{\mathrm{T}} \Delta Z_n^k \end{vmatrix}}{|U_k^{\mathrm{T}} \Delta Z_n^k|}.$$

From this expression, we see that Theorem 1.2 also holds for this transformation.

2. Theoretical results

Let us now see what happens if the sequence transformations described in Section 1 are applied to a sequence (x_n) of the form

$$x_n = x + Z_n \alpha + e_n, \quad n = 0, 1, \ldots, \tag{10}$$

where (e_n) is a vector sequence.

Such a question was already treated in [6] for the E-algorithm, and in [18] for the VTT and the BVTT. We will now prove that the same results also hold for the other strategies.

Theorem 2.1. *If any of the transformations described in Section 1 is applied to a sequence (x_n) satisfying (10), then $y_n^k = x + e_n^k$, $n = 0, 1, \ldots,$ where (e_n^k) is the sequence obtained by applying the transformation (with the same matrices U_k and Z_n) to the sequence (e_n).*

Moreover if $e_n = a_{k+1}z_n^{k+1} + a_{k+2}z_n^{k+2} + \cdots$, *then* $e_n^k = a_{k+1}\tilde{z}_n^{k+1} + a_{k+2}\tilde{z}_n^{k+2} + \cdots$, *where* $(\tilde{z}_n^{k+i})$ *is the sequence obtained by applying the transformation (with the same matrices* U_k *and* Z_n*) to the sequence* (z_n^{k+i}).

Proof. Let us give the proof for the E-strategy. The proofs for the other transformations are similar.

Applying the E-strategy to a sequence of the form (10) gives

$$y_n^k = x + Z_n\alpha + e_n - Z_n(U_k^{\mathrm{T}}\Delta Z_n^k)^{-1}U_k^{\mathrm{T}}\Delta X_n^k$$

with

$$\Delta X_n^k = \begin{pmatrix} \Delta x_n \\ \vdots \\ \Delta x_{n+k-1} \end{pmatrix} = \begin{pmatrix} \Delta Z_n\alpha + \Delta e_n \\ \vdots \\ \Delta Z_{n+k-1}\alpha + \Delta e_{n+k-1} \end{pmatrix} = \Delta Z_n^k\alpha + \Delta E_n^k,$$

where

$$\Delta E_n^k = \begin{pmatrix} \Delta e_n \\ \vdots \\ \Delta e_{n+k-1} \end{pmatrix}.$$

Thus

$$\begin{aligned} y_n^k &= x + Z_n\alpha + e_n - Z_n(U_k^{\mathrm{T}}\Delta Z_n^k)^{-1}U_k^{\mathrm{T}}\Delta Z_n^k\alpha \\ &\quad - Z_n(U_k^{\mathrm{T}}\Delta Z_n^k)^{-1}U_k^{\mathrm{T}}\Delta E_n^k \\ &= x + e_n^k, \end{aligned}$$

with

$$e_n^k = e_n - Z_n(U_k^{\mathrm{T}}\Delta Z_n^k)^{-1}U_k^{\mathrm{T}}\Delta E_n^k.$$

The second part of the theorem follows immediately from the linearity of the operator Δ. □

A consequence of this result is that e_n^k can be expressed by the same ratio of determinants and the same Schur complement formula as the corresponding y_n^k after replacing the x_n's by the e_n's. The same is true for the $\tilde{z}_n^{k+i}$.

We immediately see that (the proof is the same as for Theorem 8 in [18])

Corollary 2.2. *If any of the transformations described in Section* 1 *is applied to a sequence* (x_n) *converging to* x *and satisfying* (10), *if* $\|e_n\| = o(\|Z_n\alpha\|)$ *and if* $\|e_n^k\| = \mathcal{O}(\|e_n\|)$, *then* (y_n^k) *converges to* x *faster than* (x_n), *i.e.* $\|y_n^k - x\| = o(\|x_n - x\|)$.

We will see now that y_n^{k+m} can be expressed directly from y_n^k. Let us set $Z_n = [z_n^1, \ldots, z_n^k]$ as before and, now, $Z_n' = [z_n^{k+1}, \ldots, z_n^{k+m}]$ where $m \geqslant 1$.

For all strategies described above, y_n^{k+m} can be written as $y_n^{k+m} = \det M_n^{k+m} / \det D_n^{k+m}$ with

$$M_n^{k+m} = \begin{pmatrix} x_n & Z_n & Z'_n \\ V_n & D_n & D'_n \\ V'_n & D''_n & D'''_n \end{pmatrix}, \qquad D_n^{k+m} = \begin{pmatrix} D_n & D'_n \\ D''_n & D'''_n \end{pmatrix}.$$

We remind that

$$y_n^k = \begin{vmatrix} x_n & Z_n \\ V_n & D_n \end{vmatrix} \Big/ \det D_n.$$

By Lemma 1.1, we have

$$y_n^k = x_n - [Z_n, Z'_n] \begin{pmatrix} D_n & D'_n \\ D''_n & D'''_n \end{pmatrix}^{-1} \begin{pmatrix} V_n \\ V'_n \end{pmatrix}. \tag{11}$$

Let Y_n and Y_{n+1} be the solutions of the systems $D_n Y_n = V_n$ and $D_n^{k+m} Y_{n+1} = (V_n^{\mathrm{T}}, (V'_n)^{\mathrm{T}})^{\mathrm{T}}$, respectively. The block bordering method [19] allows to express Y_{n+1} in terms of Y_n as

$$Y_{n+1} = \begin{pmatrix} Y_n \\ 0 \end{pmatrix} + \begin{pmatrix} -D_n^{-1} D'_n \\ I \end{pmatrix} \widetilde{D}_n^{-1} (V'_n - D''_n Y_n)$$

with $\widetilde{D}_n = D'''_n - D''_n D_n^{-1} D'_n$. It follows from (11)

$$y_n^{k+m} = y_n^k + (Z_n D_n^{-1} D'_n - Z'_n) \widetilde{D}^{-1} (V'_n - D''_n Y_n)$$

since $y_n^k = x_n - Z_n D_n^{-1} V_n = x_n - Z_n Y_n$.

Let $\tilde{z}_n^{k+i}$ be the vectors obtained by applying the same transformation (that is with the same matrices U_k and Z_n) to z_n^{k+i} for $i = 1, \ldots, m$ and let $\widetilde{Z}_n = [\tilde{z}_n^{k+1}, \ldots, \tilde{z}_n^{k+m}]$. We have $\widetilde{Z}_n = Z'_n - Z_n D_n^{-1} D'_n$. Let $\widetilde{Y}_n$ be the solution of the system $(D'''_n - d''_n D_n^{-1} D'_n) \widetilde{Y}_n = V'_n - D''_n Y_n = V'_n - D''_n D_n^{-1} V_n$.

Let us remark that

- D'_n is similar to D_n but applied to the vectors z_n^{k+i} instead of z_n^i,
- D''_n is similar to D_n but with V'_n instead of V_n,
- D'''_n is similar to D_n but applied to the vectors z_n^{k+i} instead of z_n^i and with V'_n instead of V_n.

Thus, $\tilde{D}_n = D'''_n - D''_n D_n^{-1} D'_n$ is the matrix replacing D_n when the same transformation is applied to the vectors z_n^{k+i} with V'_n instead of V_n. Similarly, $\tilde{V}_n = V'_n - D''_n D_n^{-1} V_n$ is the vector replacing V_n when V'_n is used instead of V_n and we have

$$y_n^{k+m} = y_n^k - \widetilde{Z}_n \widetilde{D}_n^{-1} \widetilde{V}_n.$$

$\widetilde{Z}_n$, $\widetilde{D}_n$ and $\widetilde{V}_n$ have the same expressions as Z_n, D_n and V_n, respectively, after replacing the vectors z_n^i by the vectors $\tilde{z}_n^{k+i}$ and V_n by V'_n.

This result extends a result given in [7, 11] for the E-algorithm. It allows to compute directly y_n^{k+m} from y_n^k without computing the intermediate steps, a look-ahead procedure avoiding possible breakdowns (division by zero) or near-breakdowns (division by a number close to zero). Since, usually, the value of m is small the computation of $\widetilde{D}_n$ and the solution of a system of linear equations with this matrix is easy and does not require a recursive algorithm. Such an algorithm exists in the case of the E-algorithm [11].

3. Connection to interpolation and projection

All the ratios of determinants given in Section 1 have a common feature: in each column of the numerators, the scalars from the second row to the last one make use of the operator Δ applied to the sequence of vectors appearing in the same column of the first row. We will suppress this operator by working in a set bigger than $\mathbb{R}^p$. The denominators, which are the lower right corners of the numerators, will be treated in the same way. Owing to such manipulations, all the vector sequence transformations of Section 1 could be related to the general interpolation problem as described in [26] (see also [12]) and to projections [14].

Let us begin by the simplest case which corresponds to the Θ strategy. We define the following vectors of $\mathbb{R}^{2p}$, $v_i=(-u_i^{\mathrm{T}},u_i^{\mathrm{T}})^{\mathrm{T}}$, $s_n=(x_n^{\mathrm{T}},x_{n+1}^{\mathrm{T}})^{\mathrm{T}}$ and $r_n^i=(z_n^{i^{\mathrm{T}}},z_{n+1}^{i^{\mathrm{T}}})^{\mathrm{T}}$. Obviously, we have $(u_i,\Delta x_n)=(v_i,s_n)$ and $(u_i,\Delta z_n^j)=(v_i,r_n^j)$. Owing to this property, the ratio of determinants (8) of the Θ strategy is identical to the first p components of the vector $Y_n^k\in\mathbb{R}^{2p}$ defined by

$$Y_n^k=\begin{vmatrix} s_n & r_n^1 & \cdots & r_n^k\\ (v_1,s_n) & (v_1,r_n^1) & \cdots & (v_1,r_n^k)\\ \vdots & \vdots & & \vdots\\ (v_k,s_n) & (v_k,r_n^1) & \cdots & (v_k,r_n^k)\end{vmatrix}\Bigg/\begin{vmatrix} (v_1,r_n^1) & \cdots & (v_1,r_n^k)\\ \vdots & & \vdots\\ (v_k,r_n^1) & \cdots & (v_k,r_n^k)\end{vmatrix}. \tag{12}$$

The ratios of determinants (12) are exactly those involved in the *Recursive Projection Algorithm* (RPA) [9]. For a fixed value of n, they can be recursively computed by the rules of this algorithm

$$Y_n^k=Y_n^{k-1}-\frac{(v_k,Y_n^{k-1})}{(v_k,g^{k-1,k})}g^{k-1,k},$$
$$g^{k,i}=g^{k-1,i}-\frac{(v_k,g^{k-1,i})}{(v_k,g^{k-1,k})}g^{k-1,k},\quad i>k,$$

with $Y_n^0=s_n$ and $g^{0,i}=r_n^i, i\geqslant 1$.

The vectors $g^{k,i}$ can be expressed by a ratio of determinants similar to (12) after replacing, in the first column of the numerator, s_n by r_n^i.

Setting $r_n^0=s_n$, the vectors Y_n^k can also be computed by the *Compact Recursive Projection Algorithm* (CRPA) [9] whose rule is

$$e_k^{(i)}=e_{k-1}^{(i)}-\frac{(v_k,e_{k-1}^{(i)})}{(v_k,e_{k-1}^{(i+1)})}e_{k-1}^{(i+1)}$$

with $e_0^{(i)}=r_n^i$ and we get $e_k^{(0)}=Y_n^k$.

The vectors Y_n^k given by (12) are related to the solution of the general interpolation problem as described in [26, 12]. Indeed, let us set

$$P_k=-\begin{vmatrix} 0 & r_n^1 & \cdots & r_n^k\\ w_1 & (v_1,r_n^1) & \cdots & (v_1,r_n^k)\\ \vdots & \vdots & & \vdots\\ w_k & (v_k,r_n^1) & \cdots & (v_k,r_n^k)\end{vmatrix}\Bigg/\begin{vmatrix} (v_1,r_n^1) & \cdots & (v_1,r_n^k)\\ \vdots & & \vdots\\ (v_k,r_n^1) & \cdots & (v_k,r_n^k)\end{vmatrix},$$

where the w_i are numbers. Then P_k is the only element in $R_k=$ span$(r_n^1,\ldots,r_n^k)$ solving the general interpolation problem

$$(v_i,P_k)=w_i \quad \text{for } i=1,\ldots,k.$$

In our case, $\forall i, w_i=(v_i,s_n)$. Then, the preceding interpolation conditions can be written as

$$(v_i,P_k-s_n)=0, \quad i=1,\ldots,k$$

which shows that P_k is the oblique projection of x on R_k along $V_k^{\perp}$ where $V_k=\text{span}(v_1,\ldots,v_k)$. Moreover, since $Y_n^k=s_n-P_k$, we have $(v_i,Y_n^k)=0$ for $i=1,\ldots,k$.

Obviously, the rules of the RPA (and the CRPA) can be modified for computing recursively the vectors P_k. Such algorithms are called RIA and CRIA, respectively, where I stands for *interpolation*.

Let us now consider the case of the E-strategy. We define the vectors of $\mathbb{R}^{(k+1)p}$, $v_1=(-u^{\mathrm{T}},u^{\mathrm{T}},0^{\mathrm{T}},\ldots,0^{\mathrm{T}})^{\mathrm{T}}$, $v_2=(0^{\mathrm{T}},-u^{\mathrm{T}},u^{\mathrm{T}},0^{\mathrm{T}},\ldots,0^{\mathrm{T}})^{\mathrm{T}},\ldots,v_k=(0^{\mathrm{T}},\ldots,0^{\mathrm{T}},-u^{\mathrm{T}},u^{\mathrm{T}})^{\mathrm{T}}$, $s_n=(x_n^{\mathrm{T}},\ldots,x_{n+k}^{\mathrm{T}})^{\mathrm{T}}$ and $r_n^i=(z_n^{i^{\mathrm{T}}},\ldots,z_{n+k}^{i^{\mathrm{T}}})^{\mathrm{T}}$. Obviously, we have $(u_i,\Delta x_n)=(v_i,s_n)$ and $(u_i,\Delta z_n^j)=(v_i,r_n^j)$. Thus, the ratio of determinants (4) of the E strategy is identical to the first p components of the vector $Y_n^k\in\mathbb{R}^{(k+1)p}$ again defined by (12). It follows that the RPA and the CRPA can still be used for the recursive computation of these vectors and that similar results connecting them to interpolation and projection hold.

Equivalent results could easily be derived for the ratios of determinants obtained by the $E\Theta$ and the ΘE strategies.

4. Particular cases

We will now examine some particular cases which enter into the general framework developed in Section 1. In this Section, the vectors u and $u_1,\ldots,u_k$ do not depend on n. We will mainly examine the case where $z_n^i=\Delta x_{n+i-1}$. General results for this case will be given in the last Subsection.

4.1. The E strategy

When the vectors z_n^i are arbitrary, this strategy corresponds exactly to the vector E-algorithm [6] whose rules are (6).

For the particular choice $z_n^i=\Delta x_{n+i-1}$, the ratio of determinants (4) becomes, after some easy rearrangements,

$$y_n^k=\begin{vmatrix} x_n & \cdots & x_{n+k}\\ (u,\Delta x_n) & \cdots & (u,\Delta x_{n+k})\\ \vdots & & \vdots\\ (u,\Delta x_{n+k-1}) & \cdots & (u,\Delta x_{n+2k-1})\end{vmatrix} \Bigg/ \begin{vmatrix} 1 & \cdots & 1\\ (u,\Delta x_n) & \cdots & (u,\Delta x_{n+k})\\ \vdots & & \vdots\\ (u,\Delta x_{n+k-1}) & \cdots & (u,\Delta x_{n+2k-1})\end{vmatrix}.$$

This is the topological e-transform introduced in [3] as a generalization of Shanks' transformation. The vectors y_n^k can be recursively computed by (6) with $z_n^i=\Delta x_{n+i-1}$ but also by the topological

ε-algorithm whose rules are

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

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

with $\varepsilon_{-1}^{(n)} = 0 \in \mathbb{R}^p$, $\varepsilon_0^{(n)} = x_n$ and we obtain $\varepsilon_{2k}^{(n)} \equiv y_n^k$.

4.2. The Θ strategy

In the particular case where $z_n^i = \Delta x_{n+i-1}$, the ratio of determinants (8) becomes

$$y_n^k = \begin{vmatrix} x_n & \cdots & x_{n+k} \\ (u_1, \Delta x_n) & \cdots & (u_1, \Delta x_{n+k}) \\ \vdots & & \vdots \\ (u_k, \Delta x_n) & \cdots & (u_k, \Delta x_{n+k}) \end{vmatrix} \Bigg/ \begin{vmatrix} 1 & \cdots & 1 \\ (u_1, \Delta x_n) & \cdots & (u_1, \Delta x_{n+k}) \\ \vdots & & \vdots \\ (u_k, \Delta x_n) & \cdots & (u_k, \Delta x_{n+k}) \end{vmatrix}.$$

Such a transformation was first considered in [3]. Let β_n^k be the ratio of determinants obtained by replacing, in this expression, the first row of the numerator by $\Delta x_n, \ldots, \Delta x_{n+k}$. It has been proved by Jbilou [39] that the following recursive algorithm, called the $S\beta$-algorithm, holds

$$y_n^{k+1} = y_n^k - \frac{(u_{k+1}, \beta_n^k)}{(u_{k+1}, \beta_{n+1}^k) - (u_{k+1}, \beta_n^k)}(y_{n+1}^k - y_n^k),$$

$$\beta_n^{k+1} = \beta_n^k - \frac{(u_{k+1}, \beta_n^k)}{(u_{k+1}, \beta_{n+1}^k) - (u_{k+1}, \beta_n^k)}(\beta_{n+1}^k - \beta_n^k), \quad i > k+1$$

with $y_n^0 = x_n$ and $\beta_n^0 = \Delta x_n$.

However, this algorithm is not optimal since, although only the scalars (u_i, β_n^k) are used, the vectors β_n^k have to be stored. Setting $z_n^{k,i} = (u_i, \beta_n^k)$, we have the following cheaper recursive algorithm

$$y_n^{k+1} = y_n^k - \frac{z_n^{k,k+1}}{z_{n+1}^{k,k+1} - z_n^{k,k+1}}(y_{n+1}^k - y_n^k),$$

$$z_n^{k+1,i} = z_n^{k,i} - \frac{z_n^{k,k+1}}{z_{n+1}^{k,k+1} - z_n^{k,k+1}}(z_{n+1}^{k,i} - z_n^{k,i}), \quad i > k+1,$$

with $y_n^0 = x_n$ and $z_n^{0,i} = (u_i, \Delta x_n)$. This algorithm is exactly the H-algorithm introduced in [20]. This algorithm was built for implementing Henrici's vector sequence transformation [37, p.116] which corresponds to $k = p$, with u_i the ith vector of the canonical basis of $\mathbb{R}^p$. For this transformation, it holds

$$y_n = x_n - \Delta X_n (\Delta^2 X_n)^{-1} \Delta x_n$$

where $X_n = [x_n, \ldots, x_{n+p-1}]$. The convergence acceleration properties of this transformation were studied in [45]. More generally, the H-algorithm can be used for computing recursively, from the initial conditions $y_n^0 = x_n$ and $z_n^{0,i} = g_n^i$, ratios of determinants of the form

$$y_n^k = \begin{vmatrix} x_n & \cdots & x_{n+k} \\ g_n^1 & \cdots & g_{n+k}^1 \\ \vdots & & \vdots \\ g_n^k & \cdots & g_{n+k}^k \end{vmatrix} \Bigg/ \begin{vmatrix} 1 & \cdots & 1 \\ g_n^1 & \cdots & g_{n+k}^1 \\ \vdots & & \vdots \\ g_n^k & \cdots & g_{n+k}^k \end{vmatrix},$$

where the g_n^i are scalars.

In the case where $g_n^i = c_{n+i-1}$, the preceding ratios of determinants can be computed by the vector G-algorithm which is a generalization of the scalar one [35]

$$\left(1 - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}}\right) y_n^{k+1} = y_{n+1}^k - \frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} y_n^k$$

with $y_n^0 = x_n$. The scalars $r_k^{(n)}$ are computed by the rs-algorithm [42]

$$s_{k+1}^{(n)} = s_k^{(n+1)} \left(\frac{r_{k+1}^{(n+1)}}{r_{k+1}^{(n)}} - 1\right), \quad k, n = 0, 1, \ldots,$$

$$r_{k+1}^{(n)} = r_k^{(n+1)} \left(\frac{s_k^{(n+1)}}{s_k^{(n)}} - 1\right), \quad k = 1, 2, \ldots; \quad n = 0, 1, \ldots.$$

with $s_0^{(n)} = 1$ and $r_1^{(n)} = c_n$.

This algorithm can be considered as an intermediate between the H-algorithm where the g_n^i are arbitrary, and the topological ε-algorithm where they have fixed values.

4.3. The mixed strategies

In the case where $z_n^i = \Delta x_{n+i-1}$, the ratios of determinants can be rewritten, as for the E and the Θ strategies, with $x_n, \ldots, x_{n+k}$ as first row of the numerator, $1, \ldots, 1$ as the first row of the denominator (which has now dimension $k+1$ as the denominator), and quantities of the form $(u_i, \Delta x_{n+j})$ for the other elements. So, the ratios of determinants obtained in *all* strategies when $z_n^i = \Delta x_{n+i-1}$ enter into the general theory of reference functionals and triangular recursive schemes presented in [24]. It means, in particular, that the vectors y_n^k can be recursively computed by a triangular scheme of the form

$$y_n^k = \lambda_n^k y_n^{k-1} + \mu_n^k y_{n+1}^{k-1} \tag{13}$$

with $y_n^0 = x_n$, λ_n^k and μ_n^k being coefficients summing up to one. This scheme appears to be a generalization of the E-algorithm described above. The reciprocal of this result also holds, namely that vectors y_n^k computed by a recursive scheme similar to (13) can be expressed as a ratio of two determinants. Moreover, this theory tells us that y_n^k can be written as

$$y_n^k = \sum_{i=0}^{k} a_{ni}^k x_{n+i},$$

where the a_{ni}^k are coefficients satisfying some recurrence relationships involving λ_n^k and μ_n^k. So, these transformations are quasi-linear which means that if they are applied to the sequence $(ax_n + b)$ with $a \in \mathbb{R}$ and $b \in \mathbb{R}^p$ they produce the vectors $ay_n^k + b$. The theory of quasi-linear transformations started from an observation made by Benchiboun [2] and was developed in [10, 1].

The theory also states that these transformations can be characterized in terms of a linear functional which annihilates a certain linear space which is, in our case, the kernel N_T^k of the transformation. Such an approach was introduced by Weniger [53].

In the $E\Theta$ strategy, if $d = p$ (the dimension of the vectors) and if the vectors u_i are the vectors of the canonical basis of $\mathbb{R}^p$, we recover the generalization of Shanks transformation proposed and studied in [52]. An expression using the Schur complement and a recursive algorithm for the implementation of this transformation were also given and an application to systems of linear equations as well.

5. Applications to linear systems

One of the most important applications of vector sequence transformations is the construction and the acceleration of iterative methods for solving systems of linear equations. In this Section, we will study such an application. It is the continuation of [5, 47, 32, 13]; see [14, Sections 1.5 and 4.1] for a review.

Let us consider the $p \times p$ system $Ax = b$ and let (x_n) be an arbitrary sequence of vectors constructed by an iterative method for its solution, or related, in some way, to this system. Again, we want to construct a vector sequence transformation having N_T^k as its kernel. The main difference with the procedures described in Section 1 is that, now, for computing $a_1, \ldots, a_k$, we have the residuals $r_n = b - Ax_n$ at our disposal. If $(x_n) \in N_T^k$, then $x_n = x + Z_n\alpha$ with $Z_n = [z_n^1, \ldots, z_n^k]$ and $\alpha = (a_1, \ldots, a_k)^{\mathrm{T}}$, and it follows that x can be eliminated by using the residuals since we have

$$r_n = -AZ_n\alpha = -a_1Az_n^1 - \cdots - a_kAz_n^k. \tag{14}$$

The vector α can be computed by deriving, by the same strategies as in Section 1, a system of k linear equations in the k unknowns $a_1, \ldots, a_k$. Then, by construction, the kernel of the transformation T defined by

$$y_n = x_n - Z_n\alpha, \quad n = 0, 1, \ldots \tag{15}$$

will be N_T^k. Thus, the difference with the procedures of Section 1 is that the strategies are now applied to (14) instead of (2).

Instead of transforming the sequence (x_n) obtained by an arbitrary iterative method, the same idea can be used for the construction of iterative methods by

$$x_{n+1} = x_n - Z_n\alpha, \quad n = 0, 1, \ldots, \tag{16}$$

where α is computed by one of the strategies of Section 1.

We have the following result, if the corresponding system giving α is nonsingular

Theorem 5.1. *If $k=p$ and if, $\forall n$, the vectors $z_n^1,\dots,z_n^p$ are linearly independent, then, applying the strategies of Section 1 to* (14), *we obtain $y_n = x$ (for the transformation* (15)) *and $x_{n+1}=x$ (for the iterative method* (16)) *for $n=0,1,\dots$.*

Proof. In all strategies, the system giving $a_1,\dots,a_k$ is equivalent to $AZ_n\alpha = -r_n$. Since the vectors z_n^i are linearly independent, the matrix Z_n is nonsingular and

$$\alpha = -(AZ_n)^{-1}r_n = -Z_n^{-1}A^{-1}r_n$$

holds. Thus $y_n = x_n + Z_nZ_n^{-1}A^{-1}r_n = x_n + A^{-1}(b - Ax_n) = x_n + x - x_n = x$. Obviously, the same is true for the iterative method (16). □

We will now again examine the strategies of Section 1 when applied to the solution of linear systems. In fact, they can all be understood in the framework of formal biorthogonal polynomials [12]. It is very important to remark, by comparing (2) and (14), that, for all strategies, the systems giving $a_1,\dots,a_k$ are the same as those of Section 1 after replacing Δx_n by $-r_n$ and Δz_n^i by Az_n^i.

5.1. The E strategy

The scalars a_i are solution of the system

$$-(u, r_{n+j}) = a_1(u, Az_{n+j}^1) + \cdots + a_k(u, Az_{n+j}^k), \quad j = 0,\dots,k-1$$

and we obtain

$$y_n^k = \begin{vmatrix} x_n & z_n^1 & \cdots & z_n^k \\ -(u,r_n) & (u,Az_n^1) & \cdots & (u,Az_n^k) \\ \vdots & \vdots & & \vdots \\ -(u,r_{n+k-1}) & (u,Az_{n+k-1}^1) & \cdots & (u,Az_{n+k-1}^k) \end{vmatrix} \Bigg/ \begin{vmatrix} (u,Az_n^1) & \cdots & (u,Az_n^k) \\ \vdots & & \vdots \\ (u,Az_{n+k-1}^1) & \cdots & (u,Az_{n+k-1}^k) \end{vmatrix}. \quad (17)$$

Let (x_n) be defined by

$$x_{n+1} - x = A(x_n - x), \quad n = 0,1,\dots$$

with x_0 arbitrary. Of course, this sequence cannot be constructed in practice since x is unknown but, as we will see, this is not an obstacle. The corresponding residuals satisfy $r_{n+1} = Ar_n$, that is $r_n = A^n r_0$. Let us take, in this strategy, $z_0^i = A^{i-1}r_0$. Then

$$y_0^k = \begin{vmatrix} x_0 & r_0 & \cdots & A^{k-1}r_0 \\ -(u,r_0) & (u,Ar_0) & \cdots & (u,A^kr_0) \\ \vdots & \vdots & & \vdots \\ -(u,A^{k-1}r_0) & (u,A^kr_0) & \cdots & (u,A^{2k-1}r_0) \end{vmatrix} \Bigg/ \begin{vmatrix} (u,Ar_0) & \cdots & (u,A^kr_0) \\ \vdots & & \vdots \\ (u,A^kr_0) & \cdots & (u,A^{2k-1}r_0) \end{vmatrix} \quad (18)$$

which shows that the sequence (y_0^k) is identical to the sequence obtained by Lanczos method [40] since we have the same ratios of determinants (the ratio of determinants corresponding to Lanczos method is given, for example, in [5, p. 87ff; 22; 14, p. 161]).

In Lanczos method, $r_k = P_k(A)r_0$ where P_k is the polynomial

$$P_k(\xi) = \begin{vmatrix} 1 & \xi & \cdots & \xi^k \\ (u,r_0) & (u,Ar_0) & \cdots & (u,A^k r_0) \\ \vdots & \vdots & & \vdots \\ (u,A^{k-1}r_0) & (u,A^k r_0) & \cdots & (u,A^{2k-1}r_0) \end{vmatrix} \Bigg/ \begin{vmatrix} (u,Ar_0) & \cdots & (u,A^k r_0) \\ \vdots & & \vdots \\ (u,A^k r_0) & \cdots & (u,A^{2k-1}r_0) \end{vmatrix}. \tag{19}$$

Thus, r_k is a linear combination of the vectors $r_0,\ldots,A^k r_0$.

Lanczos-type product methods (LTPM) [17, 36] are variants of Lanczos method where the residual $r_k^* = b - Ax_k^*$ is defined by $r_k^* = P_k(A)W_k(A)r_0$ with P_k given by (19) and W_k a polynomial such that $W_k(0)=1$. Thus, setting $r'_k = W_k(A)r_0 = b - Ax'_k$, we have $r_k^* = P_k(A)r'_k$, which means that r_k^* is the linear combination of $r'_k,\ldots,A^k r'_k$ with the same coefficients as the residual of Lanczos method. So, using (19), we see that x_k^* is given by the ratio (18) after replacing in the first row, and in the first row only, x_0 by x'_k and r_0 by r'_k. So, we obtain expressions for the iterates and the residuals of any LTPM as ratios of two determinants. Such a result was first proved in [21], but by a much more complicated proof, in the case of the CGS [49] which corresponds to $W_k \equiv P_k$. Among the most well-known LTPM is the BiCGSTAB [50] where $W_k(\xi) = (1 - w_k\xi)W_{k-1}(\xi)$ with $W_0(\xi)=1$ and w_k chosen to minimize (r_k^*, r_k^*).

These ratios of determinants can also be recovered from the E strategy. In (17), let us take $r_{k+j} = A^j W_k(A)r_0 = A^j r'_k$, $z_{k+j}^i = A^{i+j-1}W_k(A)r_0 = A^{i+j-1}r'_k$ and replace u by $W_k^{-1}(A^{\mathrm{T}})u$. It is easy to see that the vectors y_k^k are identical to the vectors x_k^* obtained by a LTPM.

5.2. The Θ strategy

The system giving $a_1,\ldots,a_k$ is

$$-(u_i,r_n) = a_1(u_i,Az_n^1) + \cdots + a_k(u_i,Az_n^k), \quad i=1,\ldots,k,$$

and

$$y_n^k = \begin{vmatrix} x_n & z_n^1 & \cdots & z_n^k \\ -(u_1,r_n) & (u_1,Az_n^1) & \cdots & (u_1,Az_n^k) \\ \vdots & \vdots & & \vdots \\ -(u_k,r_n) & (u_k,Az_n^1) & \cdots & (u_k,Az_n^k) \end{vmatrix} \Bigg/ \begin{vmatrix} (u_1,Az_n^1) & \cdots & (u_1,Az_n^k) \\ \vdots & & \vdots \\ (u_k,Az_n^1) & \cdots & (u_k,Az_n^k) \end{vmatrix} \tag{20}$$

holds.

If we take $z_0^i = A^{i-1}r_0$ and $u_i = A^{\mathrm{T}^{i-1}}u$, then we recover the ratios (18), which shows that the sequence (y_0^k) produced by the Θ strategy is identical to the sequence obtained by Lanczos method. Taking $z_k^i = A^{i-1}W_k(A)r_0$ and $u_i = (A^{i-1}W_k^{-1}(A))^{\mathrm{T}}u$ in (20), it is easy to see that the vectors y_k^k are identical to the iterates x_k^* of a LTPM since $(u_i,r_k) = (u,A^{i-1}r_0)$ and $(u_i,Az_k^j) = (u,A^{i+j-1}r_0)$.

Let us now set $\rho_n = b - Ay_n$, and choose α_n in order to minimize (ρ_n,ρ_n). Since we have $\rho_n = r_n + AZ_n\alpha_n$, it follows

$$y_n = x_n + Z_n[(AZ_n)^{\mathrm{T}}AZ_n]^{-1}(AZ_n)^{\mathrm{T}}r_n.$$

The idea behind this vector sequence transformation is quite similar to the least-squares variant of the VTT described at the end of Section 1.2. It is easy to see that the kernel of this transformation, introduced in [15] and called the *multiparameter PR2 acceleration*, is the set of sequences such that $\forall n$, $x_n = x + Z_n\alpha_n$, that is the same kernel as the BVTT. Thus, in this case, we see that the fixed vector α has been replaced by a vector α_n depending on n.

5.3. *The mixed strategies*

Similar results hold for the mixed strategies. We will make use of the equation

$$-(u_i, r_{n+j}) = a_1(u_i, Az^1_{n+j}) + \cdots + a_k(u_i, Zz^k_{n+j}). \tag{21}$$

Again, let $d \geqslant 1$ be a *fixed* integer. We define m and r by $k = md + r$ with $0 \leqslant r < d$.

In the E strategy, we took $r_n = A^n r_0$ and, in the Θ strategy, $u_i = A^{\mathrm{T}^{i-1}} u$. In both cases we also had $z^i_0 = A^{i-1} r_0$. So, in a mixed strategy, it seems logical to consider again the same choices. However, as it could be seen from the determinantal expressions given below, since one row becomes

$$-(u, A^{i+j-1} r_0)\ (u, A^{i+j} r_0) \cdots (u, A^{i+j+k-1} r_0)$$

we will have identical rows in the determinants and this choice is impossible.

5.3.1. *The $E\Theta$ strategy*

In this strategy, we will write down (21) for $i=1,\ldots,d$ and $j=0,\ldots,m-1$, and also for $i=1,\ldots,r$ and $j=m$.

Solving the system, we obtain

$$y^k_n = \begin{vmatrix} x_n & z^1_n & \cdots & z^k_n \\ -(u_1, r_n) & (u_1, Az^1_n) & \cdots & (u_1, Az^k_n) \\ \vdots & \vdots & & \vdots \\ -(u_d, r_n) & (u_d, Az^1_n) & \cdots & (u_d, Az^k_n) \\ \vdots & \vdots & & \vdots \\ -(u_1, r_{n+m-1}) & (u_1, Az^1_{n+m-1}) & \cdots & (u_1, Az^k_{n+m-1}) \\ \vdots & \vdots & & \vdots \\ -(u_d, r_{n+m-1}) & (u_d, Az^1_{n+m-1}) & \cdots & (u_d, Az^k_{n+m-1}) \\ -(u_1, r_{n+m}) & (u_1, Az^1_{n+m}) & \cdots & (u_1, Az^k_{n+m}) \\ \vdots & \vdots & & \vdots \\ -(u_r, r_{n+m}) & (u_r, Az^1_{n+m}) & \cdots & (u_r, Az^k_{n+m}) \end{vmatrix} \Bigg/ \det D_n,$$

where D_n is the matrix obtained by deleting the first row and the first column of the matrix whose determinant forms the numerator.

5.3.2. The ΘE strategy

We will now use (21) for $i=1,\dots,m$ and $j=0,\dots,d-1$, and also for $i=m+1$ and $j=0,\dots,r-1$. Solving this system, we get

$$y_n^k = \begin{vmatrix} x_n & z_n^1 & \cdots & z_n^k \\ -(u_1,r_n) & (u_1,Az_n^1) & \cdots & (u_1,Az_n^k) \\ \vdots & \vdots & & \vdots \\ -(u_1,r_{n+d-1}) & (u_1,Az_{n+d-1}^1) & \cdots & (u_1,Az_{n+d-1}^k) \\ \vdots & \vdots & & \vdots \\ -(u_m,r_n) & (u_m,Az_n^1) & \cdots & (u_m,Az_n^k) \\ \vdots & \vdots & & \vdots \\ -(u_m,r_{n+d-1}) & (u_m,Az_{n+d-1}^1) & \cdots & (u_m,Az_{n+d-1}^k) \\ -(u_{m+1},r_n) & (u_{m+1},Az_n^1) & \cdots & (u_{m+1},Az_n^k) \\ \vdots & \vdots & & \vdots \\ -(u_{m+1},r_{n+r-1}) & (u_{m+1},Az_{n+r-1}^1) & \cdots & (u_{m+1},Az_{n+r-1}^k) \end{vmatrix} \Bigg/ \det D_n,$$

where D_n is the matrix obtained by deleting the first row and the first column of the matrix whose determinant forms the numerator.

5.4. A particular case

Let us set $A = M - N$ and consider the iterations

$$x_{n+1} = M^{-1}Nx_n + M^{-1}b, \quad n = 0,1,\dots$$

Let $P(\xi) = \alpha_0 + \alpha_1\xi + \cdots + \alpha_m\xi^m$ be the minimal polynomial of $M^{-1}N$ for the vector $x_0 - x$ with $m \leqslant p$. By the Cayley–Hamilton theorem, we have (see, for example, [16, p.303]), $\forall n$, $\alpha_0(x_n - x) + \cdots + \alpha_m(x_{n+m} - x) = 0$ with, since $I - M^{-1}N$ is nonsingular, $\alpha_0 + \cdots + \alpha_m \neq 0$. This relation can be rewritten as $x_n = x + a_1\Delta x_n + \cdots + a_m\Delta x_{n+m-1}$ which shows that $(x_n) \in N_T^k$ for the choice $z_n^i = \Delta x_{n+i-1}$.

It is easy to check that

$$\Delta x_n = M^{-1}(b - Ax_n) = r_n,$$

$$r_{n+1} = M^{-1}Nr_n,$$

$$\Delta^i r_n = (-1)^i (M^{-1}A)^i r_n.$$

r_n, as defined above, is the residual of the left preconditioned system $M^{-1}Ax = M^{-1}b$.

In the previous subsections, we made use of (14). We will now again apply our strategies to (2) with the choice $z_n^i = \Delta x_{n+i-1} = r_{n+i-1}$.

So, the first (vector) row of the numerator and the scalar rows of the numerator and the denominator become

$$\begin{matrix} x_n & r_n & \cdots & r_{n+k-1} \\ (u_i, r_{n+j}) & (u_i, \Delta r_{n+j}) & \cdots & (u_i, \Delta r_{n+j+k-1}) \end{matrix} \tag{22}$$

with u instead of u_i for the E strategy.

Replacing each column, from the third one, by its difference with the preceding one leads, for the first row of the numerator and the scalar rows, to

$$\begin{matrix} x_n & r_n & \Delta r_n & \cdots & \Delta^{k-1} r_n \\ (u_i, r_{n+j}) & (u_i, \Delta r_{n+j}) & (u_i, \Delta^2 r_{n+j}) & \cdots & (u_i, \Delta^k r_{n+j}) \end{matrix}$$

that is

$$\begin{matrix} x_n & r_n & -(M^{-1}A)r_n & \cdots & (-1)^{k-1}(M^{-1}A)^{k-1} r_n \\ (u_i, r_{n+j}) & -(u_i, (M^{-1}A) r_{n+j}) & (u_i, (M^{-1}A)^2 r_{n+j}) & \cdots & (-1)^k (u_i, (M^{-1}A)^k r_{n+j}). \end{matrix}$$

After multiplying some rows and columns by -1, we finally get

$$\begin{matrix} x_n & r_n & (M^{-1}A)r_n & \cdots & (M^{-1}A)^{k-1} r_n \\ -(u_i, r_{n+j}) & (u_i, (M^{-1}A) r_{n+j}) & (u_i, (M^{-1}A)^2 r_{n+j}) & \cdots & (u_i, (M^{-1}A)^k r_{n+j}). \end{matrix}$$

So, for the E strategy and for the Θ strategy with the choice $u_i = (AM^{-T})^{i-1} u$, we immediately see that the vectors y_0^k are again those of Lanczos method (left preconditioned by M^{-1}).

If the vectors u_i are arbitrary (but, of course, linearly independent), we obtain the MMPE [3, 41] for which the following result was proved in [41].

Theorem 5.2. *If the eigenvectors of $M^{-1}N$ are linearly independent and if its eigenvalues satisfy $|\lambda_1| \geqslant |\lambda_2| \geqslant \cdots |\lambda_k| > |\lambda_{k+1}| \geqslant \cdots$, then $\|y_n^k - x\| = \mathcal{O}(|\lambda_{k+1}|^n)$.*

A similar result holds for the topological ε-algorithm [4, 46, 48].

When $k = 1$ and $u_1 = r_n$, the *steepest descent acceleration* described in [14, p.128] is recovered. The choice $u_i = \Delta x_{n+i-1} = r_{n+i-1}$ leads to the MPE [25], while, if $u_i = \Delta^2 x_{n+i-1} = -M^{-1} A r_{n+i}$, we recover the RRE [28], all these methods being left preconditioned by M^{-1}. As explained in [13], other choices can be interesting. For example, the choice $u_i = (M^{-1}A)^{i-1} r_0$ leads to the method GCR [43] which is mathematically equivalent to GMRES [44], while taking $z_n^i = \Delta^2 x_{n+i-1}$ and $u_i = \Delta^q x_{n+i-1}$ corresponds to a method proposed by Germain-Bonne [33].

For the E strategy, let us, in the numerator and in the denominator, replace the second scalar row by its difference with the previous one, and so on. These rows become

$$-(u, \Delta^j r_n)\ (u, (M^{-1}A)\Delta^j r_n)\ (u, (M^{-1}A)^k \Delta^j r_n), \quad j = 0, \ldots, k-1.$$

After multiplying some rows and columns by -1, we obtain, for y_0^k, the same ratio of determinants as in Lanczos method (left preconditioned by M^{-1}). Due to the connection, described in Section 4.1, between the E strategy and the topological ε-algorithm, we recover a result, first proved in

[5, pp.84–90; 186–189], about the equivalence between Lanczos method and the topological ε-algorithm. These results (some of them more or less already given in [5, pp.182–186]) were exploited in [34] for deriving a transpose-free version of the algorithm Lanczos/Orthomin and giving a look-around algorithm for the treatment of breakdowns and near-breakdowns. Obviously, the $E\Theta$ and the ΘE strategies, as well as other choices for the vectors u_i, lead to variants of Lanczos method such as that described in [14, pp.175,176] or that based on vector Padé approximants [51]. Such variants are under consideration.

Let us come back to the expressions (22) for the rows of the determinants. They can be written as

$$\begin{matrix} x_n & \Delta x_n & \cdots & \Delta x_{n+k-1} \\ (u_i, \Delta x_{n+j}) & (u_i, \Delta^2 x_{n+j}) & \cdots & (u_i, \Delta^2 x_{n+j+k-1}). \end{matrix}$$

Replacing each column by its sum with the previous one, we obtain

$$\begin{matrix} x_n & \cdots & x_{n+k} \\ (u_i, \Delta x_{n+j}) & \cdots & (u_i, \Delta x_{n+j+k-1}) \end{matrix}$$

for the numerator and the same determinant for the denominator after replacing the first row by $1 \cdots 1$.

For the E strategy, we recover the topological e-transform (that is the topological ε-algorithm). For the Θ strategy, we obtain the ratio of determinants involved in the $S\beta$-algorithm [39].

Remark 5.3. Exactly the same transformations are obtained by applying our strategies to (14) with $z_n^i = (-1)^{i-1}(M^{-1}A)^{i-1}r_n$.

5.5. Some numerical examples

The aim of this section is to illustrate the steepest descent acceleration which consists of transforming a sequence (x_n) obtained by an arbitrary iterative method for solving $Ax = b$ into the sequence (y_n) given

$$y_n = x_n + \lambda_n r_n,$$

$$\rho_n = M^{-1}(b - Ay_n) = r_n - \lambda_n M^{-1}Ar_n,$$

with $r_n = M^{-1}(b - Ax_n)$, $u_n = r_n$ and $\lambda_n = (u_n, r_n)/(u_n, M^{-1}Ar_n)$.

In all examples, the solution is randomly chosen and the right-hand side is computed accordingly. The curves in solid lines are the Euclidean norms of r_n, while those in dashed lines are the norms of ρ_n.

Let (x_n) be computed by the method of Jacobi from $x_0 = 0$. It is well known that it converges, for example, if A is strictly diagonally dominant.

In Fig. 1, the matrix is `grcar(50)+triw(50)` from [38]. Its condition number is $2.9\ 10^{10}$.

In Fig. 2, the matrix is `redheff(100)+10*eye(100)` from [38]. Its condition number is 10.03.

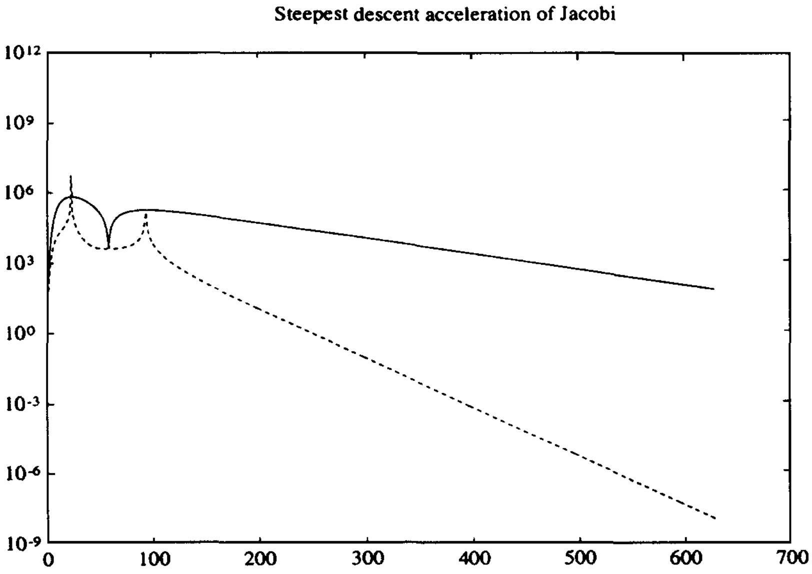

Fig. 1. Example 1.

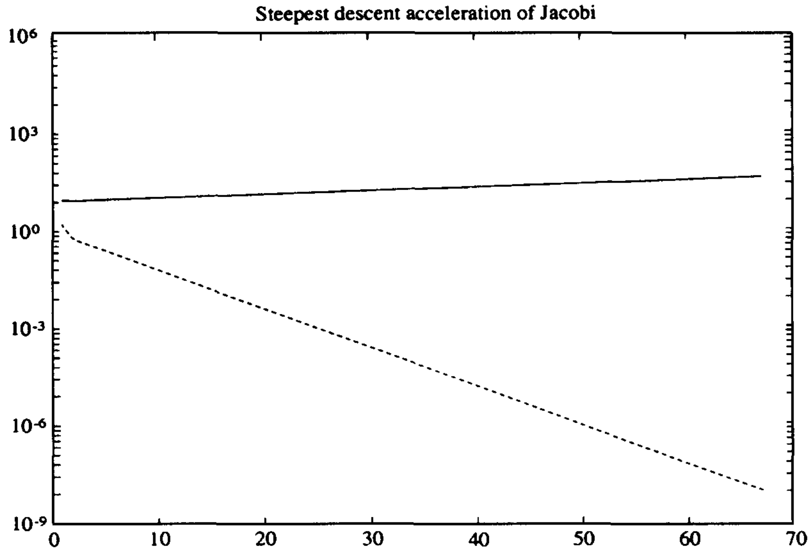

Fig. 2. Example 2.

In Fig. 3, the matrix is `dramadah(100)+4*eye(100)` from [38]. Its condition number is 13.15.

In Fig. 4, the matrix is `lehmer(100)+100*eye(100)` from [38]. Its condition number is 1.55.

So, we see that, as stated by the theoretical results [41], the convergence is accelerated and that, in some cases, the steepest descent acceleration can even transform a diverging sequence into a converging one. Similar results hold for the topological and the vector ε-algorithms [4].

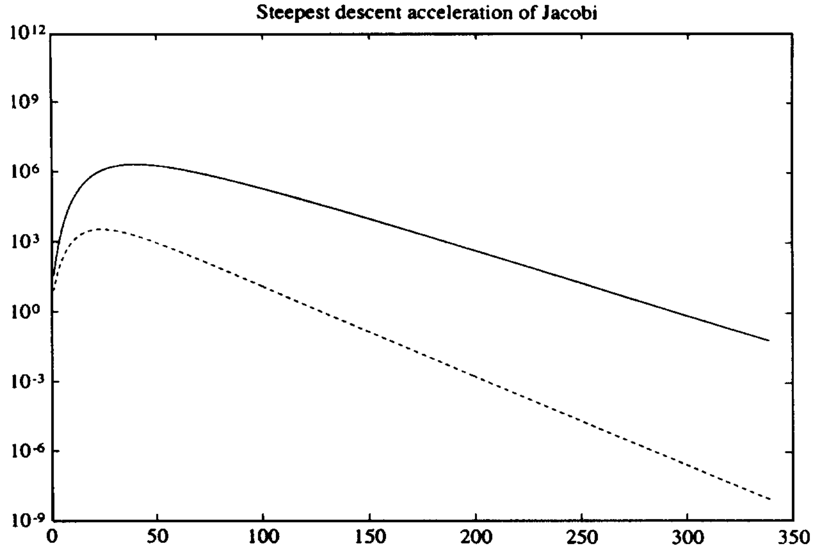

Fig. 3. Example 3.

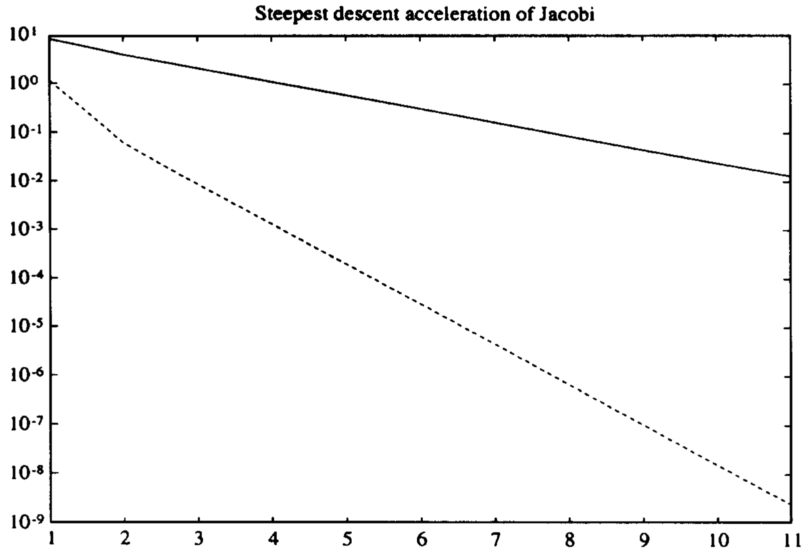

Fig. 4. Example 4.

Almost identical results were obtained with $u_n = r_0$ and also with the transformation

$$y_n = x_n + \lambda_n r_n,$$

$$\rho_n = b - Ay_n = r_n - \lambda_n A r_n,$$

with $r_n = b - Ax_n$, $u_n = r_n$ and $\lambda_n = (u_n, r_n)/(u_n, Ar_n)$.

The steepest descent acceleration was also tried on Lanczos/Orthomin with various matrices, but none of the examples provides interesting results.

References

[1] A. Benazzouz, Quasilinear sequence transformations, Numer. Algorithms 15 (1997) 275–285.
[2] M.D. Benchiboun, Some results on non-linear and non-logarithmic sequences and their acceleration, Zastosow. Mat. 21 (1992) 407–425.
[3] C. Brezinski, Généralisation de la transformation de Shanks, de la table de Padé et de l'ε-algorithme, Calcolo 12 (1975) 317–360.
[4] C. Brezinski, Computation of the eigenelements of a matrix by the ε-algorithm, Linear Algebra Appl. 11 (1975) 7–20.
[5] C. Brezinski, Padé-Type Approximation and General Orthogonal Polynomials, Birkhäuser, Basel, 1980.
[6] C. Brezinski, A general extrapolation algorithm, Numer. Math. 35 (1980) 175–187.
[7] C. Brezinski, The Mühlbach–Neville–Aitken algorithms and some extensions, BIT 20 (1980) 444–451.
[8] C. Brezinski, Some new convergence acceleration methods, Math. Comput. 39 (1982) 133–145.
[9] C. Brezinski, Recursive interpolation, extrapolation and projection, J. Comput. Appl. Math. 9 (1983) 369–376.
[10] C. Brezinski, Quasi-linear extrapolation processes, in: R.P. Agarwal et al. (Eds.), Numerical Mathematics; Singapore 1988, Birkhäuser, Basel, 1988, pp. 61–78.
[11] 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.
[12] C. Brezinski, Biorthogonality and its Applications to Numerical Analysis, Marcel Dekker, New York, 1992.
[13] C. Brezinski, Biorthogonality and conjugate gradient-type algorithms, in: R.P. Agarwal (Ed.), Contributions in Numerical Mathematics, World Scientific, Singapore, 1993, pp. 55–70.
[14] C. Brezinski, Projection Methods for Systems of Equations, North-Holland, Amsterdam, 1997.
[15] C. Brezinski, J.P. Chehab, Multiparameter iterative schemes for the solution of systems of linear and nonlinear equations, SIAM J. Sci. Comput., to appear.
[16] C. Brezinski, M. Redivo-Zaglia, Extrapolation Methods. Theory and Practice, North-Holland, Amsterdam, 1991.
[17] C. Brezinski, M. Redivo-Zaglia, Look-ahead in BiCGSTAB and other product methods for linear systems, BIT 35 (1995) 169–201.
[18] C. Brezinski, M. Redivo-Zaglia, Vector and matrix sequence transformations based on biorthogonality, Appl. Numer. Math. 21 (1996) 353–373.
[19] C. Brezinski, M. Redivo-Zaglia, H. Sadok, A breakdown-free Lanczos type algorithm for solving linear systems, Numer. Math. 63 (1992) 29–38.
[20] C. Brezinski, H. Sadok, Vector sequence transformations and fixed point methods, in: C. Taylor et al. (Eds.), Numerical Methods in Laminar and Turbulent Flows, Pineridge Press, Swansea, 1987, pp. 3–11.
[21] C. Brezinski, H. Sadok, Some vector sequence transformations with applications to systems of equations, Numer. Algorithms 3 (1992) 75–80.
[22] C. Brezinski, H. Sadok, Lanczos type algorithms for solving systems of linear equations, Appl. Numer. Math. 11 (1993) 443–473.
[23] C. Brezinski, A. Salam, Matrix and vector sequence transformations revisited, Proc. Edinb. Math. Soc. 38 (1995) 495–510.
[24] C. Brezinski, G. Walz, Sequences of transformations and triangular recursion schemes, with applications in numerical analysis, J. Comput. Appl. Math. 34 (1991) 361–383.
[25] S. Cabay, L.W. Jackson, A polynomial extrapolation method for finding limits and antilimits of vector sequences, SIAM J. Numer. Anal. 13 (1976) 734–752.
[26] P.J. Davis, Interpolation and Approximation, Dover, New York, 1975.
[27] W.J. Duncan, Some devices for the solution of large sets of simultaneous linear equations, Philos. Mag. Ser. 7 35 (1944) 660–670.
[28] R.P. Eddy, Extrapolation to the limit of a vector sequence, in: P.C.C. Wang (Ed.), Information Linkage Between Applied Mathematics and Industry, Academic Press, New York, 1979, pp. 387–396.

[29] V.N. Faddeeva, Computational Methods of Linear Algebra, Dover, New York, 1959.
[30] D.K. Faddeev, V.N. Faddeeva, Computational Methods of Linear Algebra, W.H. Freeman and Co., San Francisco, 1963.
[31] W.F. Ford, A. Sidi, Recursive algorithms for vector extrapolation methods, Appl. Numer. Math. 4 (1988) 477–489.
[32] W. Gander, G.H. Golub, D. Gruntz, Solving linear equations by extrapolation, in: J.S. Kovalik (Ed.), Supercomputing, Springer, Berlin, 1989, pp. 279–293.
[33] B. Germain-Bonne, Estimation de la Limite de Suites et Formalisation de Procédés d'Accélération de Convergence, Thèse, Université des Sciences et Technologies de Lille, 1978.
[34] P.R. Graves-Morris, A "Look-around Lanczos" algorithm for solving a system of linear equations, Numer. Algorithms 15 (1997) 247–274.
[35] H.L. Gray, T.A. Atchison, G.V. McWilliams, Higher order *G*-transformations, SIAM J. Numer. Anal. 8 (1971) 365–381.
[36] M.H. Gutknecht, Variants of BiCGStab for matrices with complex spectrum, SIAM J. Sci. Comput. 14 (1993) 1020–1033.
[37] P. Henrici, Elements of Numerical Analysis, Wiley, New York, 1964.
[38] N.J. Higham, The test matrix toolbox for MATLAB (Version 3.0), Numerical Analysis Report No. 276, Departments of Mathematics, The University of Manchester, September 1995.
[39] K. Jbilou, A general projection algorithm for solving systems of linear equations, Numer. Algorithms 4 (1993) 361–377.
[40] C. Lanczos, Solution of systems of linear equations by minimized iterations, J. Res. Natl. Bur. Stand. 49 (1952) 33–53.
[41] B.P. Pugachev, Acceleration of the convergence of iterative processes and a method for solving systems of nonlinear equations, Comput. Math. Math. Phys. 17 (5) (1978) 199–207.
[42] W.C. Pye, T.A. Atchison, An algorithm for the computation of the higher order *G*-transformation, SIAM J. Numer. Anal. 10 (1973) 1–10.
[43] Y. Saad, M.H. Schultz, Conjugate gradient-like algorithms for solving nonsymmetric linear systems, Math. Comput. 44 (1985) 417–424.
[44] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Statist. Comput. 7 (1986) 856–869.
[45] H. Sadok, About Henrici's transformation for accelerating vector sequences, J. Comput. Appl. Math. 29 (1990) 101–110.
[46] A. Sidi, Convergence and stability properties of minimal polynomial and reduced rank extrapolation algorithms, SIAM J. Numer. Anal. 23 (1986) 197–209.
[47] A. Sidi, Extrapolation vs. projection methods for linear systems of equations, J. Comput. Appl. Math. 22 (1988) 71–88.
[48] A. Sidi, J. Bridger, Convergence and stability analyses of some vector extrapolation methods in the presence of defective iteration matrices, J. Comput. Appl. Math. 22 (1988) 35–61.
[49] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 10 (1989) 36–52.
[50] H.A. Van der Vorst, Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 13 (1992) 631–644.
[51] J. Van Iseghem, Vector Padé approximants, in: R. Vichnevetsky, J. Vignes (Eds.), Numerical Mathematics and Applications, North-Holland, Amsterdam, 1985, pp. 73–77.
[52] J. Van Iseghem, Convergence of vectorial sequences. Applications, Numer. Math. 68 (1994) 549–562.
[53] E.J. Weniger, Nonlinear sequence transformations for the acceleration of convergence and the summation of divergent series, Comput. Phys. Rep. 10 (1989) 189–371.
[54] J. Wimp, Sequence Transformations and their Applications, Academic Press, New York, 1981.
[55] P. Wynn, Acceleration techniques for iterated vector and matrix problems, Math. Comput. 16 (1962) 301–322.