Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

MATRIX SHANKS TRANSFORMATIONS*

CLAUDE BREZINSKI† AND MICHELA REDIVO-ZAGLIA‡

Abstract. Shanks' transformation is a well know sequence transformation for accelerating the convergence of scalar sequences. It has been extended to the case of sequences of vectors and sequences of square matrices satisfying a linear difference equation with scalar coefficients. In this paper, a more general extension to the matrix case where the matrices can be rectangular and satisfy a difference equation with matrix coefficients is proposed and studied. In the particular case of square matrices, the new transformation can be recursively implemented by the matrix ε-algorithm of Wynn. Then, the transformation is related to matrix Padé-type and Padé approximants. Numerical experiments showing the interest of this transformation end the paper.

Key words. Acceleration methods, Sequence transformations, Shanks transformation, Padé approximation.

AMS subject classifications. 65B05, 65B99, 65F60, 65H10, 41A21.

1. Introduction to Shanks transformations.

Let (s_n) be a sequence of elements of a vector space E which converges to s. If the convergence is slow, it is possible to transform it into a new sequence of elements of E (or a set of new sequences) which, under some assumptions, converges faster to the same limit. Such *sequence transformations*, also known as *extrapolation methods*, have been widely studied and have a bunch of applications: Systems of linear and nonlinear equations [8], matrix equations [7, 20], matrix functions [8], block linear systems [23], nonlinear Fredholm integral equations [9], ...

Among these transformations for accelerating the convergence of scalar sequences, an important one is due to Shanks [36, 37]. It can be recursively implemented by the scalar ε-algorithm of Wynn [39] who also extended it to sequences of vectors and matrices [40]. Due to its complicated algebraic theory, which involved Clifford algebra, another extension of the transformation to sequences of elements of a general vector space was proposed and studied in [2]. This transformation can be implemented by the topological ε-algorithm (TEA) which was recently greatly simplified [7], and whose software and applications were also published [8, 9, 12]. An extension of the theory was then proposed in [10]. Let us give an account of it, sufficient for understanding the case of matrix sequences treated in this paper.

We make use of the forward difference operator Δ defined by $\Delta u_n = u_{n+1} - u_n$ where (u_n) is any sequence. When applied to sequences with two indexes, Δ operates on the lower index. Powers of Δ are recursively given by $\Delta^k = \Delta(\Delta^{k-1})$ where Δ^0 is the identity.

The idea behind Shanks transformation is to construct a set of sequence transformations $(s_n) \longmapsto \{(t_n^{(k)})\}$ such that, for a fixed value of k and for all n, $t_n^{(k)} = s$ if, for all n,

$$\alpha_0(s_n - s) + \cdots + \alpha_k(s_{n+k} - s) = 0 \in E, \tag{1.1}$$

*Received by the editors on November 5, 2018. Accepted for publication on May 24, 2019. Handling Editor: Dario Bini. Corresponding Author: Michela Redivo-Zaglia.

†Laboratoire Paul Painlevé, UMR CNRS 8524, UFR de Mathématiques, Université de Lille, 59655-Villeneuve d'Ascq cedex, France (Claude.Brezinski@univ-lille.fr).

‡Università degli Studi di Padova, Dipartimento di Matematica "Tullio Levi-Civita", Via Trieste 63, 35121-Padova, Italy (Michela.RedivoZaglia@unipd.it).

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

where the *scalars* α_0 and α_k are such that $\alpha_0\alpha_k \neq 0$. It does not restrict the generality to assume that $\alpha_0 + \cdots + \alpha_k = 1$, thus leading to the transformation

$$t_n^{(k)} = \alpha_0 s_n + \cdots + \alpha_k s_{n+k}, \quad n = 0, 1, \ldots, \tag{1.2}$$

which, if the coefficients α_i can be computed, possesses the property that, $\forall n, t_n^{(k)} = s$ for sequences satisfying (1.1). The set of these sequences is called the *kernel* of the transformation. This can be achieved if a sequence (t_n) of elements of E satisfying, $\forall n$,

$$\alpha_0 t_n + \cdots + \alpha_k t_{n+k} = 0 \in E, \tag{1.3}$$

is known. In order to compute the α_i's, this relation in the vector space E has to be transformed into k scalar equations. This is obtained by considering its duality product with an element y of E^*, the algebraic dual space of E, that is the space of linear functionals on it, which gives

$$\alpha_0\langle y, t_n\rangle + \cdots + \alpha_k\langle y, t_{n+k}\rangle = 0 \in \mathbb{R}. \tag{1.4}$$

Together with the condition that the α_i's sum up to 1, k such equations have to be considered in order to obtain a system of $k+1$ equations whose solution is $\alpha_0, \ldots, \alpha_k$. Then, $t_n^{(k)}$ can be computed by (1.2), and we obtain $\forall n, t_n^{(k)} = s$. This linear system can be written and solved even if the sequence (s_n) does not satisfy (1.1), thus producing the sequence transformation defined by (1.2). Let us mention that (1.2) and (1.3) can be respectively written as

$$\begin{aligned} t_n^{(k)} &= s_n - \beta_1\Delta s_n - \cdots - \beta_k\Delta s_{n+k-1} \\ 0 &= t_n - \beta_1\Delta t_n - \cdots - \beta_k\Delta t_{n+k-1} \end{aligned}$$

with $\beta_i = -(\alpha_i + \cdots + \alpha_k)$ for $i = 1, \ldots, k$. Notice that this new formulation implicitly assumes that the α_is sum up to one. In the particular case where E is $\mathbb{R}^p$, the duality product becomes the usual inner product that is $\langle y, t_n\rangle = y^T t_n$.

Remark 1.1. It can also be of interest to fix the index n, and to consider the sequence $(t_n^{(k)})_k$. In Example 4, we consider such a sequence. We see that the gain brought is not sufficient to justify the large increase in the computational and storage costs. However, the contrary can also occurs in certain cases.

There are three strategies for writing the linear system giving the α_i's thus leading to the transformations called *Coupled Topological Shanks Transformations*. A complete framework is provided in [10]. In all the strategies, $t_n^{(k)}$ can be written as a ratio of two determinants. The determinants in the numerators of these expressions are elements of E. They are the linear combinations of the elements of E of their first rows which are obtained by computing them by the classical rule for expanding a determinant with respect to a row. Thus, so is $t_n^{(k)}$ since the denominators are scalars (see below). Moreover, using the extended Schur determinantal formula [10], $t_n^{(k)}$ can be expressed as an extended Schur complement.

We are considering here only the first two strategies since, in the matrix case, the third one can be recovered as a particular case of the first one.

Remark 1.2. Many sequence transformations are defined as a ratio of determinants, and/or as the solution of a linear system of equations, and/or by a recursive algorithm. A common assumption is to assume that the determinants in the denominators are different from zero, and/or that the linear systems are nonsingular, and/or that there is no division by zero in the recursive algorithms. These assumptions are made in the sequel.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

These strategies for determining the α_i's are:

- *Polynomial extrapolation strategy.*
  In (1.4), n is fixed, and we write it for k linearly independent linear functionals $y_i \in E^*$ which can depend on n. We have

$$t_n^{(k)} = \frac{\begin{vmatrix} s_n & \cdots & s_{n+k} \\ \langle y_1, t_n\rangle & \cdots & \langle y_1, t_{n+k}\rangle \\ \vdots & & \vdots \\ \langle y_k, t_n\rangle & \cdots & \langle y_k, t_{n+k}\rangle \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ \langle y_1, t_n\rangle & \cdots & \langle y_1, t_{n+k}\rangle \\ \vdots & & \vdots \\ \langle y_k, t_n\rangle & \cdots & \langle y_k, t_{n+k}\rangle \end{vmatrix}} = \frac{\begin{vmatrix} s_n & \Delta s_n & \cdots & \Delta s_{n+k-1} \\ \langle y_1, t_n\rangle & \langle y_1, \Delta t_n\rangle & \cdots & \langle y_1, \Delta t_{n+k-1}\rangle \\ \vdots & \vdots & & \vdots \\ \langle y_k, t_n\rangle & \langle y_k, \Delta t_n\rangle & \cdots & \langle y_k, \Delta t_{n+k-1}\rangle \end{vmatrix}}{\begin{vmatrix} \langle y_1, \Delta t_n\rangle & \cdots & \langle y_1, \Delta t_{n+k-1}\rangle \\ \vdots & & \vdots \\ \langle y_k, \Delta t_n\rangle & \cdots & \langle y_k, \Delta t_{n+k-1}\rangle \end{vmatrix}}$$

$$= s_n - [\Delta s_n, \ldots, s_{n+k-1}] \begin{pmatrix} \langle y_1, \Delta t_n\rangle & \cdots & \langle y_1, \Delta t_{n+k-1}\rangle \\ \vdots & & \vdots \\ \langle y_k, \Delta t_n\rangle & \cdots & \langle y_k, \Delta t_{n+k-1}\rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle y_1, t_n\rangle \\ \vdots \\ \langle y_k, t_n\rangle \end{pmatrix}. \tag{1.5}$$

- *Shanks strategy.*
  In (1.4), $y \in E^*$ is fixed, and we write it for the indexes $n, \ldots, n+k-1$. We have

$$t_n^{(k)} = \frac{\begin{vmatrix} s_n & \cdots & s_{n+k} \\ \langle y, t_n\rangle & \cdots & \langle y, t_{n+k}\rangle \\ \vdots & & \vdots \\ \langle y, t_{n+k-1}\rangle & \cdots & \langle y, t_{n+2k-1}\rangle \end{vmatrix}}{\begin{vmatrix} 1 & \cdots & 1 \\ \langle y, t_n\rangle & \cdots & \langle y, t_{n+k}\rangle \\ \vdots & & \vdots \\ \langle y, t_{n+k-1}\rangle & \cdots & \langle y, t_{n+2k-1}\rangle \end{vmatrix}} = \frac{\begin{vmatrix} s_n & \Delta s_n & \cdots & \Delta s_{n+k-1} \\ \langle y, t_n\rangle & \langle y, \Delta t_n\rangle & \cdots & \langle y, \Delta t_{n+k-1}\rangle \\ \vdots & \vdots & & \vdots \\ \langle y, t_{n+k-1}\rangle & \langle y, \Delta t_{n+k-1}\rangle & \cdots & \langle y, \Delta t_{n+2k-2}\rangle \end{vmatrix}}{\begin{vmatrix} \langle y, \Delta t_n\rangle & \cdots & \langle y, \Delta t_{n+k-1}\rangle \\ \vdots & & \vdots \\ \langle y, \Delta t_{n+k-1}\rangle & \cdots & \langle y, \Delta t_{n+2k-2}\rangle \end{vmatrix}}$$

$$= s_n - [\Delta s_n, \ldots, s_{n+k-1}] \begin{pmatrix} \langle y, \Delta t_n\rangle & \cdots & \langle y, \Delta t_{n+k-1}\rangle \\ \vdots & & \vdots \\ \langle y, \Delta t_{n+k-1}\rangle & \cdots & \langle y_k, \Delta t_{n+2k-2}\rangle \end{pmatrix}^{-1} \begin{pmatrix} \langle y, t_n\rangle \\ \vdots \\ \langle y, t_{n+k-1}\rangle \end{pmatrix}. \tag{1.6}$$

2. Rectangular matrix Shanks transformations. We will now show how to construct Shanks transformations for sequences of rectangular matrices. The interest of our approach is mostly theoretical. It allows to compare the expressions of Shanks transformations with others that are of interest and have been recently studied (see, for example, [10]). Their implementation by the formulæ given below or by those of [29] requires the solution of block linear systems or the inversion of block matrices, and they can only be used on examples of relatively small dimensions. However, they show their effectiveness. In the case of square matrices, treated in Section 3, the matrix ε-algorithm presented in Section 4 is used in the applications given in Section 6.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

Matrices are either in bold or denoted by capital letters. The square identity and the zero matrices of dimension m are denoted by $\mathbf{I}_m$ and $\mathbf{0}_m$ respectively, and the rectangular ones of dimension $m \times q$ by $\mathbf{I}_{m\times q}$ and $\mathbf{0}_{m\times q}$. The products between matrices have to be understood as multiplications by blocks.

In these generalizations, the $\boldsymbol{\alpha}_i$'s and the $\boldsymbol{\beta}_i$'s are real square matrices instead of scalars, and the $\mathbf{s}_n$'s and the $\mathbf{t}_n$'s are real rectangular matrices (the extension to the complex case is obvious). Our approach is related, but different, to those presented in [24], [25] and [26].

Our starting point is to assume that the sequence $(\mathbf{s}_n)$ satisfies a recurrence relation of a form similar to (1.1), and that a coupled relation of a form similar to (1.3) holds for the sequence $(\mathbf{t}_n)$. Then, strategies similar to those used above for sequences of elements of a vector space E are used but the duality product between the matrices $\mathbf{y}$ and $\mathbf{t}_n$ is replaced by the matrix product $\mathbf{y}^T\mathbf{t}_n$. However, according to the side of the matrix products, we have left and right transformations.

2.1. Left transformations. We consider the matrix difference equation of order k where $\mathbf{s}_n, \mathbf{s} \in \mathbb{R}^{p\times s}$

$$(\mathbf{s}_n - \mathbf{s})\boldsymbol{\alpha}_0 + \cdots + (\mathbf{s}_{n+k} - \mathbf{s})\boldsymbol{\alpha}_k = \mathbf{0}_{p\times s}, \quad \forall n, \tag{2.7}$$

with $\boldsymbol{\alpha}_i \in \mathbb{R}^{s\times s}$, and where the matrices $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_k$ are nonsingular. This relation can be equivalently written as

$$\mathbf{s}_n - \mathbf{s} = \Delta\mathbf{s}_n\boldsymbol{\beta}_1 + \cdots + \Delta\mathbf{s}_{n+k-1}\boldsymbol{\beta}_k, \quad \forall n,$$

with $\boldsymbol{\beta}_i = -(\boldsymbol{\alpha}_i + \cdots + \boldsymbol{\alpha}_k)$ for $i = 1, \ldots, k$.

Obviously the matrix $\boldsymbol{\alpha}_0 + \cdots + \boldsymbol{\alpha}_k$ has to be nonsingular, and, as in the scalar case, since all the $\boldsymbol{\alpha}_i$'s can be multiplied by a common nonsingular matrix, it does not restrict the generality if we assume that

$$\boldsymbol{\alpha}_0 + \cdots + \boldsymbol{\alpha}_k = \mathbf{I}_s \in \mathbb{R}^{s\times s}. \tag{2.8}$$

Finally, assume that there exists a known sequence $(\mathbf{t}_n)$ of matrices in $\mathbb{R}^{p\times s}$ such that,

$$\mathbf{t}_n\boldsymbol{\alpha}_0 + \cdots + \mathbf{t}_{n+k}\boldsymbol{\alpha}_k = \mathbf{0}_{p\times s}, \quad \forall n. \tag{2.9}$$

For defining a matrix generalization of Shanks transformation, we have to write down a system of $k+1$ matrix equations of dimension $s \times s$ for computing the matrices $\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_k$. Then, the left matrix sequence transformation $(\mathbf{s}_n) \longmapsto (\mathbf{t}_n^{(k)})$ is defined by

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n\boldsymbol{\alpha}_0 + \cdots + \mathbf{s}_{n+k}\boldsymbol{\alpha}_k \in \mathbb{R}^{p\times s}. \tag{2.10}$$

The transformation can also be written as

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n - \sum_{i=1}^{k} \Delta\mathbf{s}_{n+i-1}\boldsymbol{\beta}_i, \tag{2.11}$$

where $\boldsymbol{\beta}_i = -(\boldsymbol{\alpha}_i + \cdots + \boldsymbol{\alpha}_k) \in \mathbb{R}^{s\times s}$, $i = 1, \ldots, k$. Similarly, (2.9) can also be written as

$$\mathbf{t}_n - (\Delta\mathbf{t}_n\boldsymbol{\beta}_1 + \cdots + \Delta\mathbf{t}_{n+k-1}\boldsymbol{\beta}_k) = \mathbf{0}_{p\times s}. \tag{2.12}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

The definition of the transformation either by (2.10) or by (2.11) needs the knowledge of the $s \times s$ matrices $\boldsymbol{\alpha}_i$'s or $\boldsymbol{\beta}_i$'s. If $p \neq s$ it is not possible to compute them directly since the number of unknowns and the number of equations that can be written is different. Thus, a different procedure has to be used. As in the case of a general vector space E, the linear system for the matrix coefficients $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ can be obtained by the two following strategies

- *Polynomial extrapolation strategy.*
  We write (2.9) only for the index n, and multiply it on the left by k linearly independent matrices $\mathbf{y}_1^T, \ldots, \mathbf{y}_k^T \in \mathbb{R}^{s\times p}$, which leads to

$$\mathbf{y}_i^T \mathbf{t}_n \boldsymbol{\alpha}_0 + \cdots + \mathbf{y}_i^T \mathbf{t}_{n+k} \boldsymbol{\alpha}_k = \mathbf{0}_s, \quad i = 1, \ldots, k.$$

  Adding the normalization condition (2.8), we obtain the $\boldsymbol{\alpha}_i$'s as the solution of a matrix system of linear equations, and the following formula follows from (2.10)

$$\mathbf{t}_n^{(k)} = [\mathbf{s}_n, \ldots, \mathbf{s}_{n+k}] \begin{pmatrix} \mathbf{I}_s & \cdots & \mathbf{I}_s \\ \mathbf{y}_1^T \mathbf{t}_n & \cdots & \mathbf{y}_1^T \mathbf{t}_{n+k} \\ \vdots & & \vdots \\ \mathbf{y}_k^T \mathbf{t}_n & \cdots & \mathbf{y}_k^T \mathbf{t}_{n+k} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{I}_s \\ \mathbf{0}_s \\ \vdots \\ \mathbf{0}_s \end{pmatrix}. \tag{2.13}$$

  Similarly, from (2.12), the system giving the $\boldsymbol{\beta}_i$'s is

$$\mathbf{y}_i^T \Delta\mathbf{t}_n \boldsymbol{\beta}_1 + \cdots + \mathbf{y}_i^T \Delta\mathbf{t}_{n+k-1} \boldsymbol{\beta}_k = \mathbf{y}_i^T \mathbf{t}_n, \quad i = 1, \ldots, k.$$

  Let $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_k] \in \mathbb{R}^{p\times ks}$, $T_n^{(k)} = [\mathbf{t}_n, \ldots, \mathbf{t}_{n+k-1}] \in \mathbb{R}^{p\times ks}$, and $\mathbf{t}_{n,1}^{(k)}$ be the first block of the matrix $T_n^{(k)}$, that is $\mathbf{t}_{n,1}^{(k)} = \mathbf{t}_n \in \mathbb{R}^{p\times s}$. Then $\boldsymbol{\beta}^T = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_k^T)^T = (Y^T \Delta T_n^{(k)})^{-1} Y^T \mathbf{t}_{n,1}^{(k)}$, and we get from (2.11)

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n - [\Delta\mathbf{s}_n, \ldots, \Delta\mathbf{s}_{n+k-1}](Y^T \Delta T_n^{(k)})^{-1} Y^T \mathbf{t}_{n,1}^{(k)}, \tag{2.14}$$

  Our notation means that $(Y^T \Delta T_n^{(k)}) \in \mathbb{R}^{ks\times ks}$ is the block matrix whose elements are $\mathbf{y}_i^T \Delta\mathbf{t}_{n+j-1}$ for $i, j = 1, \ldots, k$. Also $Y^T \mathbf{t}_{n,1}^{(k)} \in \mathbb{R}^{ks\times s}$ with elements $\mathbf{y}_i^T \mathbf{t}_n$ for $i = 1, \ldots, k$, and it follows that $(Y^T \Delta T_n^{(k)})^{-1} Y^T \mathbf{t}_{n,1}^{(k)}$ is a $ks \times s$ matrix, and $\mathbf{t}_n^{(k)} \in \mathbb{R}^{p\times s}$.
  This transformation is a particular case of the so-called *Matrix Generalized Recursive Projection Algorithm (MGRPA)* defined and studied by Messaoudi [29]. It can be recursively implemented by his *Algorithm A6*, a generalization to the matrix case of an algorithm given in [28] which itself generalizes the *Recursive Projection Algorithm (RPA)* proposed in [4]. Related algorithms are also given in [23] and [25] where new Schur complement identities are proved. Let us mention that determinants of block matrices, whose blocks are square and have the same dimension, can be computed by block Gaussian elimination (see, for example, [32]).
  When $\mathbf{t}_n = \Delta\mathbf{s}_n$, and according to the choice of the matrices $\mathbf{y}_i$, we obtain matrix generalizations of well-known vector sequence transformations:
  - the *Modified Minimal Polynomial Extrapolation (MMPE)* [2, 33] for matrices $\mathbf{y}_i$ which do not depend on n,
  - the *Minimal Polynomial Extrapolation (MPE)* [11] for $\mathbf{y}_i = \Delta\mathbf{s}_{n+i-1}$,
  - the *Reduced Rank Extrapolation (RRE)* [16, 27] for $\mathbf{y}_i = \Delta^2\mathbf{s}_{n+i-1}$.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

- *Shanks strategy.*
  We write (2.9) for the indexes $n, \ldots, n+k-1$, and multiply each of these relations on the left by a matrix $\mathbf{y}^T \in \mathbb{R}^{s\times p}$. We thus obtain

$$\mathbf{y}^T\mathbf{t}_{n+i}\boldsymbol{\alpha}_0 + \cdots + \mathbf{y}^T\mathbf{t}_{n+k+i}\boldsymbol{\alpha}_k = \mathbf{0}_s, \quad i = 0, \ldots, k-1,$$

  which, together with the normalization condition (2.8) furnishes a matrix linear system for the $\boldsymbol{\alpha}_i$'s. Thus, again from (2.10), we have

$$\mathbf{t}_n^{(k)} = [\mathbf{s}_n, \ldots, \mathbf{s}_{n+k}] \begin{pmatrix} \mathbf{I}_s & \cdots & \mathbf{I}_s \\ \mathbf{y}^T\mathbf{t}_n & \cdots & \mathbf{y}^T\mathbf{t}_{n+k} \\ \vdots & & \vdots \\ \mathbf{y}^T\mathbf{t}_{n+k-1} & \cdots & \mathbf{y}^T\mathbf{t}_{n+2k-1} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{I}_s \\ \mathbf{0}_s \\ \vdots \\ \mathbf{0}_s \end{pmatrix}. \tag{2.15}$$

  For the $\boldsymbol{\beta}_i$'s, we have the system

$$\mathbf{y}^T\Delta\mathbf{t}_{n+i}\boldsymbol{\beta}_1 + \cdots + \mathbf{y}^T\Delta\mathbf{t}_{n+k+i-1}\boldsymbol{\beta}_k = \mathbf{y}^T\mathbf{t}_{n+i}, \quad i = 0, \ldots, k-1.$$

  Thus, $\boldsymbol{\beta}^T = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_k^T)^T = (Y^T\Delta T_n^{(k)})^{-1}Y^T\mathbf{t}_{n,1}^{(k)}$, and, from (2.11),

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n - [\Delta\mathbf{s}_n, \ldots, \Delta\mathbf{s}_{n+k-1}](Y^T\Delta T_n^{(k)})^{-1}Y^T\mathbf{t}_{n,1}^{(k)}, \tag{2.16}$$

  with now

(2.17)
$$Y = \begin{pmatrix} \mathbf{y} & \mathbf{0}_{p\times s} & \cdots & \mathbf{0}_{p\times s} \\ \mathbf{0}_{p\times s} & \mathbf{y} & \cdots & \mathbf{0}_{p\times s} \\ \vdots & \vdots & & \vdots \\ \mathbf{0}_{p\times s} & \mathbf{0}_{p\times s} & \cdots & \mathbf{y} \end{pmatrix} \in \mathbb{R}^{kp\times ks} \text{ and } T_n^{(k)} = \begin{pmatrix} \mathbf{t}_n & \mathbf{t}_{n+1} & \cdots & \mathbf{t}_{n+k-1} \\ \mathbf{t}_{n+1} & \mathbf{t}_{n+2} & \cdots & \mathbf{t}_{n+k} \\ \vdots & \vdots & & \vdots \\ \mathbf{t}_{n+k-1} & \mathbf{t}_{n+k} & \cdots & \mathbf{t}_{n+2k-2} \end{pmatrix} \in \mathbb{R}^{kp\times ks},$$

  $\mathbf{y} \in \mathbb{R}^{p\times s}$ on the diagonal of Y, and where $\mathbf{t}_{n,1}^k \in \mathbb{R}^{kp\times s}$ is the first column of the matrix $T_n^{(k)}$ defined in (2.17). Thus, $(Y^T\Delta T_n^{(k)}) \in \mathbb{R}^{ks\times ks}$, $Y^T\mathbf{t}_{n,1}^{(k)} \in \mathbb{R}^{ks\times s}$, and it follows that $(Y^T\Delta T_n^{(k)})^{-1}Y^T\mathbf{t}_{n,1}^{(k)} \in \mathbb{R}^{ks\times s}$, and $\mathbf{t}_n^{(k)} \in \mathbb{R}^{p\times s}$.
  When $\mathbf{t}_n = \Delta\mathbf{s}_n$, this transformation generalizes to the matrix case the topological Shanks transformation (also known as the *Topological Epsilon Algorithm (TEA)*) introduced in [2]. This transformation is also a particular case of the so-called *Matrix E-Algorithm* defined and studied by Messaoudi [29], and which can be recursively implemented by his *Algorithm A4*, a generalization of the scalar and vector E-algorithm [3].

Thus, for these two left matrix transformations, we have, by construction:

THEOREM 2.1. *If the sequence* $(\mathbf{s}_n)$ *satisfies* (2.7)*, then, for all* n, $\mathbf{t}_n^{(k)} = \mathbf{s}$.

These left matrix Shanks transformations were also studied by Jbilou and Messaoudi [24] where quite similar results can be found but by a different approach. These authors also gave an efficient way for implementing the matrix MMPE. Techniques similar to those used by Sidi [38] can also be extended to the matrix case.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

2.2. Right transformations. We now start from the matrix difference equation

$$\boldsymbol{\alpha}_0(\mathbf{s}_n - \mathbf{s}) + \cdots + \boldsymbol{\alpha}_k(\mathbf{s}_{n+k} - \mathbf{s}) = \mathbf{0}_{q\times r}, \quad \forall n, \tag{2.18}$$

where $\mathbf{s}_n, \mathbf{s} \in \mathbb{R}^{q\times r}$ and $\boldsymbol{\alpha}_i \in \mathbb{R}^{q\times q}$, and where the matrices $\boldsymbol{\alpha}_0$ and $\boldsymbol{\alpha}_k$ are nonsingular. Equivalently, this relation can be written as

$$\mathbf{s}_n - \mathbf{s} = \boldsymbol{\beta}_1\Delta\mathbf{s}_n + \cdots + \boldsymbol{\beta}_k\Delta\mathbf{s}_{n+k-1},$$

with $\boldsymbol{\beta}_i = -(\boldsymbol{\alpha}_i + \cdots + \boldsymbol{\alpha}_k)$ for $i = 1, \ldots, k$.

Without any restriction to the generality, we assume again the normalization condition

$$\boldsymbol{\alpha}_0 + \cdots + \boldsymbol{\alpha}_k = \mathbf{I}_q. \tag{2.19}$$

We also assume that the sequence $(\mathbf{t}_n)$ of matrices in $\mathbb{R}^{q\times r}$ satisfies,

$$\boldsymbol{\alpha}_0\mathbf{t}_n + \cdots + \boldsymbol{\alpha}_k\mathbf{t}_{n+k} = \mathbf{0}_{q\times r}, \quad \forall n. \tag{2.20}$$

We have to write down a system of $k+1$ matrix equations of dimension $q \times q$ for obtaining the matrices $\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_k$. Then, the right matrix sequence transformation $(\mathbf{s}_n) \longmapsto (\mathbf{t}_n^{(k)})$ is defined by

$$\mathbf{t}_n^{(k)} = \boldsymbol{\alpha}_0\mathbf{s}_n + \cdots + \boldsymbol{\alpha}_k\mathbf{s}_{n+k}. \tag{2.21}$$

As in the left case, the transformation can also be written as

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n - \sum_{i=1}^{k} \boldsymbol{\beta}_i\Delta\mathbf{s}_{n+i-1},$$

where $\boldsymbol{\beta}_i = -(\boldsymbol{\alpha}_i + \cdots + \boldsymbol{\alpha}_k) \in \mathbb{R}^{q\times q}$, $i = 1, \ldots, k$. Similarly, for (2.20), we have

$$\mathbf{t}_n - (\boldsymbol{\beta}_1\Delta\mathbf{t}_n + \cdots + \boldsymbol{\beta}_k\Delta\mathbf{t}_{n+k-1}) = \mathbf{0}_{q\times r}.$$

As in the left case, if $p \neq s$, it is impossible to obtain directly a number of equations equal to the number of unknowns for computing the matrix coefficients $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$. Again, such a system can be obtained by two different ways

- *Polynomial extrapolation strategy.*
  We write (2.20) only for the index n, and multiply it on the right by $\mathbf{y}_1^T, \ldots, \mathbf{y}_k^T \in \mathbb{R}^{r\times q}$, which leads to
  $$\boldsymbol{\alpha}_0\mathbf{t}_n\mathbf{y}_i^T + \cdots + \boldsymbol{\alpha}_k\mathbf{t}_{n+k}\mathbf{y}_i^T = \mathbf{0}_q, \quad i = 1, \ldots, k.$$
  Adding the normalization condition (2.19), we obtain the $\boldsymbol{\alpha}_i$'s as the solution of the matrix system of linear equations,
  $$[\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_k] \begin{pmatrix} \mathbf{I}_q & \mathbf{t}_n\mathbf{y}_1^T & \cdots & \mathbf{t}_n\mathbf{y}_k^T \\ \vdots & \vdots & & \vdots \\ \mathbf{I}_q & \mathbf{t}_{n+k}\mathbf{y}_1^T & \cdots & \mathbf{t}_{n+k}\mathbf{y}_k^T \end{pmatrix} = [\mathbf{I}_q, \mathbf{0}_q, \ldots, \mathbf{0}_q],$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

and it holds

$$\mathbf{t}_n^{(k)} = [\mathbf{I}_q, \mathbf{0}_q, \ldots, \mathbf{0}_q] \begin{pmatrix} \mathbf{I}_q & \mathbf{t}_n\mathbf{y}_1^T & \cdots & \mathbf{t}_n\mathbf{y}_k^T \\ \vdots & \vdots & & \vdots \\ \mathbf{I}_q & \mathbf{t}_{n+k}\mathbf{y}_1^T & \cdots & \mathbf{t}_{n+k}\mathbf{y}_k^T \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{s}_n \\ \vdots \\ \mathbf{s}_{n+k} \end{pmatrix}. \tag{2.22}$$

The system giving the $\boldsymbol{\beta}_i$'s is now

$$\boldsymbol{\beta}_1 \Delta\mathbf{t}_n\mathbf{y}_i^T + \cdots + \boldsymbol{\beta}_k \Delta\mathbf{t}_{n+k-1}\mathbf{y}_i^T = \mathbf{t}_n\mathbf{y}_i^T, \quad i = 1, \ldots, k.$$

Therefore, we have the following extended Schur complement formula

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n - \mathbf{t}_n[\mathbf{y}_1^T, \ldots, \mathbf{y}_k^T] \left(\begin{pmatrix} \Delta\mathbf{t}_n \\ \vdots \\ \Delta\mathbf{t}_{n+k-1} \end{pmatrix} [\mathbf{y}_1^T, \ldots, \mathbf{y}_k^T] \right)^{-1} \begin{pmatrix} \Delta\mathbf{s}_n \\ \vdots \\ \Delta\mathbf{s}_{n+k-1} \end{pmatrix}. \tag{2.23}$$

This formula has a structure quite similar to that of (2.14). Indeed, set

$$\widetilde{Y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_k \end{pmatrix} \in \mathbb{R}^{kq\times r}, \quad \widetilde{T}_n^{(k)} = \begin{pmatrix} \mathbf{t}_n \\ \vdots \\ \mathbf{t}_{n+k-1} \end{pmatrix} \in \mathbb{R}^{kq\times r},$$

that is, $\widetilde{Y}^T = [\mathbf{y}_1^T, \ldots, \mathbf{y}_k^T] \in \mathbb{R}^{r\times kq}$ and $(\widetilde{T}_n^{(k)})^T = [\mathbf{t}_n^T, \ldots, \mathbf{t}_{n+k-1}^T] \in \mathbb{R}^{r\times kq}$, and let $\mathbf{t}_{n,1}^{(k)}$ be the block matrix in the first row of the matrix $\widetilde{T}_n^{(k)}$, that is $\mathbf{t}_{n,1}^{(k)} = \mathbf{t}_n \in \mathbb{R}^{q\times r}$. Then $\boldsymbol{\beta}^T = (\boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_k^T)^T = \mathbf{t}_{n,1}^{(k)}\widetilde{Y}^T(\Delta\widetilde{T}_n^{(k)}\widetilde{Y}^T)^{-1} \in \mathbb{R}^{q\times kq}$. Therefore, we have the following extended Schur complement formula

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n - \mathbf{t}_{n,1}^{(k)}\widetilde{Y}^T(\Delta\widetilde{T}_n^{(k)}\widetilde{Y}^T)^{-1} \begin{pmatrix} \Delta\mathbf{s}_n \\ \vdots \\ \Delta\mathbf{s}_{n+k-1} \end{pmatrix}.$$

According to the choice of the matrices $\mathbf{y}_i$, we obtain the following particular cases of the right matrix Shanks transformation when $\mathbf{t}_n = \Delta\mathbf{s}_n$:

- the *Modified Minimal Polynomial Extrapolation (MMPE)* [2, 33] for matrices $\mathbf{y}_i$ which do not depend on n,
- the *Minimal Polynomial Extrapolation (MPE)* [11] for $\mathbf{y}_i = \Delta\mathbf{s}_{n+i-1}$,
- the *Reduced Rank Extrapolation (RRE)* [16, 27] for $\mathbf{y}_i = \Delta^2\mathbf{s}_{n+i-1}$.

- *Shanks strategy.*
We write (2.20) for the indexes $n, \ldots, n+k-1$, and multiply each of these relations on the right by a matrix $\mathbf{y}^T \in \mathbb{R}^{r\times q}$. We thus obtain

$$\boldsymbol{\alpha}_0\mathbf{t}_{n+i}\mathbf{y}^T + \cdots + \boldsymbol{\alpha}_k\mathbf{t}_{n+k+i}\mathbf{y}^T = \mathbf{0}_p, \quad i = 0, \ldots, k-1,$$

which, together with the normalization condition furnishes a matrix linear system for the $\boldsymbol{\alpha}_i$'s. Thus, we have the system

$$[\boldsymbol{\alpha}_0, \ldots, \boldsymbol{\alpha}_k] \begin{pmatrix} \mathbf{I}_q & \mathbf{t}_n\mathbf{y}^T & \cdots & \mathbf{t}_{n+k-1}\mathbf{y}^T \\ \vdots & \vdots & & \vdots \\ \mathbf{I}_q & \mathbf{t}_{n+k}\mathbf{y}^T & \cdots & \mathbf{t}_{n+2k-1}\mathbf{y}^T \end{pmatrix} = [\mathbf{I}_q, \mathbf{0}_q, \ldots, \mathbf{0}_q],$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

and it follows

$$\mathbf{t}_n^{(k)} = [\mathbf{I}_q, \mathbf{0}_q, \ldots, \mathbf{0}_q] \begin{pmatrix} \mathbf{I}_q & \mathbf{t}_n\mathbf{y}^T & \cdots & \mathbf{t}_{n+k-1}\mathbf{y}^T \\ \vdots & \vdots & & \vdots \\ \mathbf{I}_q & \mathbf{t}_{n+k}\mathbf{y}^T & \cdots & \mathbf{t}_{n+2k-1}\mathbf{y}^T \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{s}_n \\ \vdots \\ \mathbf{s}_{n+k} \end{pmatrix}. \tag{2.24}$$

This transformation is another generalization of the topological Shanks transformation when $\mathbf{t}_n = \Delta\mathbf{s}_n$.
For this strategy, the system for the $\boldsymbol{\beta}_i$'s is

$$\boldsymbol{\beta}_1\Delta\mathbf{t}_{n+i}\mathbf{y}^T + \cdots + \boldsymbol{\beta}_k\Delta\mathbf{t}_{n+k+i-1}\mathbf{y}^T = \mathbf{t}_{n+i}\mathbf{y}^T, \quad i = 0, \ldots, k-1,$$

and the extended Schur complement formula follows

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n - [\mathbf{t}_n, \ldots, \mathbf{t}_{n+k-1}]\mathbf{y}^T \left(\Delta T_n^{(k)} Y^T\right)^{-1} \begin{pmatrix} \Delta\mathbf{s}_n \\ \vdots \\ \Delta\mathbf{s}_{n+k-1} \end{pmatrix}, \tag{2.25}$$

where Y and $T_n^{(k)}$ have the same structure as in (2.17).

Thus, for these two right matrix transformations, we have, by construction:

THEOREM 2.2. *If the sequence* $(\mathbf{s}_n)$ *satisfies* (2.18)*, then, for all* n, $\mathbf{t}_n^{(k)} = \mathbf{s}$.

2.3. Transposition left-right. If $p = r$ and $s = q$, the left and right cases are transposed one from each other. Indeed, transposing the expression (2.7) of the kernel of the left transformation gives

$$\boldsymbol{\alpha}_0^T(\mathbf{s}_n^T - \mathbf{s}^T) + \cdots + \boldsymbol{\alpha}_k^T(\mathbf{s}_{n+k}^T - \mathbf{s}^T) = \mathbf{0}_{q\times r}.$$

Similarly, transposing (2.9), we get

$$\boldsymbol{\alpha}_0^T\mathbf{t}_n^T + \cdots + \boldsymbol{\alpha}_k^T\mathbf{t}_{n+k}^T = \mathbf{0}_{q\times r},$$

and the transformation becomes

$$\begin{aligned}(\mathbf{t}_n^{(k)})^T &= \boldsymbol{\alpha}_0^T\mathbf{s}_n^T + \cdots + \boldsymbol{\alpha}_k^T\mathbf{s}_{n+k}^T \\ &= \mathbf{s}_n^T - \sum_{i=1}^{k} \boldsymbol{\beta}_i^T\Delta\mathbf{s}_{n+i-1}^T.\end{aligned}$$

If we transpose (2.13) and (2.14), we obtain respectively, after also transposing the $\mathbf{y}_i$'s,

$$\begin{aligned}(\mathbf{t}_n^{(k)})^T &= [\mathbf{I}_q, \mathbf{0}_q, \ldots, \mathbf{0}_q] \begin{pmatrix} \mathbf{I}_q & \mathbf{t}_n^T\mathbf{y}_1 & \cdots & \mathbf{t}_n^T\mathbf{y}_k \\ \vdots & \vdots & & \vdots \\ \mathbf{I}_q & \mathbf{t}_{n+k}^T\mathbf{y}_1 & \cdots & \mathbf{t}_{n+k}^T\mathbf{y}_k \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{s}_n^T \\ \vdots \\ \mathbf{s}_{n+k}^T \end{pmatrix} \\ &= \mathbf{s}_n^T - \mathbf{t}_n^T[\mathbf{y}_1, \ldots, \mathbf{y}_k] \left(\begin{pmatrix} \Delta\mathbf{t}_n^T \\ \vdots \\ \Delta\mathbf{t}_{n+k-1}^T \end{pmatrix} [\mathbf{y}_1, \ldots, \mathbf{y}_k] \right)^{-1} \begin{pmatrix} \Delta\mathbf{s}_n^T \\ \vdots \\ \Delta\mathbf{s}_{n+k-1}^T \end{pmatrix},\end{aligned}$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

which shows that (2.22) and (2.23) are recovered.

Similar results are obtained by transposing the formulae of the right Shanks strategy. Thus, the right transformations can be obtained by applying the left ones to $(\mathbf{s}_n^T)$ and $(\mathbf{t}_n^T)$, after replacing the matrices $\mathbf{y}_i$'s and $\mathbf{y}$ by their transposes, and transposing the results $\mathbf{t}_n^{(k)}$, and conversely.

3. Square matrix Shanks transformations. Let us now consider the case where the matrices $\mathbf{s}_n$ and $\mathbf{t}_n$ are square, that is $p = s$. It is now possible to directly write down a system with the same number of unknowns and equations, and the matrix Y appearing in the polynomial extrapolation strategy and in the Shanks strategy is no longer needed.

In the left case, the matrices $\boldsymbol{\alpha}_i$ are solution of the system

$$\begin{array}{ccccccc}
\boldsymbol{\alpha}_0 & + & \cdots & + & \boldsymbol{\alpha}_k & = & \mathbf{I}_p \\
\mathbf{t}_n\boldsymbol{\alpha}_0 & + & \cdots & + & \mathbf{t}_{n+k}\boldsymbol{\alpha}_k & = & \mathbf{0}_p \\
\vdots & & & & \vdots & & \vdots \\
\mathbf{t}_{n+k-1}\boldsymbol{\alpha}_0 & + & \cdots & + & \mathbf{t}_{n+2k-1}\boldsymbol{\alpha}_k & = & \mathbf{0}_p,
\end{array}$$

and the left matrix Shanks transformation defined by (2.10) becomes

$$\mathbf{t}_n^{(k)} = [\mathbf{s}_n, \ldots, \mathbf{s}_{n+k}] \begin{pmatrix} \mathbf{I}_p & \cdots & \mathbf{I}_p \\ \mathbf{t}_n & \cdots & \mathbf{t}_{n+k} \\ \vdots & & \vdots \\ \mathbf{t}_{n+k-1} & \cdots & \mathbf{t}_{n+2k-1} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{I}_p \\ \mathbf{0}_p \\ \vdots \\ \mathbf{0}_p \end{pmatrix}.$$

In the right case, it similarly holds from (2.21)

$$\mathbf{t}_n^{(k)} = [\mathbf{I}_p, \mathbf{0}_p, \cdots, \mathbf{0}_p] \begin{pmatrix} \mathbf{I}_p & \mathbf{t}_n & \cdots & \mathbf{t}_{n+k-1} \\ \vdots & \vdots & & \vdots \\ \mathbf{I}_p & \mathbf{t}_{n+k} & \cdots & \mathbf{t}_{n+2k-1} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{s}_n \\ \vdots \\ \mathbf{s}_{n+k} \end{pmatrix}.$$

These formulae correspond respectively to (2.15) and (2.24) with $\mathbf{y} = \mathbf{I}$ of the corresponding dimension. Similarly, (2.16) and (2.25) lead to

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n - [\Delta\mathbf{s}_n, \ldots, \Delta\mathbf{s}_{n+k-1}](\Delta T_n^{(k)})^{-1} \begin{pmatrix} \mathbf{t}_n \\ \vdots \\ \mathbf{t}_{n+k-1} \end{pmatrix},$$

and to

$$\mathbf{t}_n^{(k)} = \mathbf{s}_n - [\mathbf{t}_n, \ldots, \mathbf{t}_{n+k-1}] \left(\Delta T_n^{(k)}\right)^{-1} \begin{pmatrix} \Delta\mathbf{s}_n \\ \vdots \\ \Delta\mathbf{s}_{n+k-1} \end{pmatrix},$$

with $T_n^{(k)}$ defined as in (2.17).

We immediately see that these two formulae coincide when $\mathbf{t}_n = \Delta\mathbf{s}_n$. In this case, following the original notation used by Shanks [37], we set $\mathbf{e}_k(\mathbf{s}_n) = \mathbf{t}_n^{(k)}$, and we can express the transformation as a Schur

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

complement

$$\mathbf{e}_k(\mathbf{s}_n) = (M^{(k+1)}(\mathbf{s}_n)/\Delta^2 S_n^{(k)}),$$

where

$$M^{(k+1)}(\mathbf{s}_n) = \begin{pmatrix} \mathbf{s}_n & \Delta\mathbf{s}_n & \cdots & \Delta\mathbf{s}_{n+k-1} \\ \Delta\mathbf{s}_n & \Delta^2\mathbf{s}_n & \cdots & \Delta^2\mathbf{s}_{n+k-1} \\ \vdots & \vdots & & \vdots \\ \Delta\mathbf{s}_{n+k-1} & \Delta^2\mathbf{s}_{n+k-1} & \cdots & \Delta^2\mathbf{s}_{n+2k-2} \end{pmatrix}, \quad S_n^{(k)} = \begin{pmatrix} \mathbf{s}_n & \cdots & \mathbf{s}_{n+k-1} \\ \vdots & & \vdots \\ \mathbf{s}_{n+k-1} & \cdots & \mathbf{s}_{n+2k-2} \end{pmatrix}.$$

Moreover, in this case, since our approach follows the same lines as that of Salam [35], his results apply and $\mathbf{e}_k(\mathbf{s}_n)$ can be expressed using designants, a generalization of determinants in a noncommutative algebra. Thus, as a consequence, these matrix transformations can be recursively implemented as we will see in the next Section 4.

4. The matrix ε-algorithm. The ε-algorithm of Wynn [39] is a recursive algorithm for implementing Shanks transformation in the scalar case. It was extended to the square matrix case by Wynn [40], and it's rule is

$$\boldsymbol{\varepsilon}_{k+1}^{(n)} = \boldsymbol{\varepsilon}_{k-1}^{(n+1)} + (\boldsymbol{\varepsilon}_k^{(n+1)} - \boldsymbol{\varepsilon}_k^{(n)})^{-1}, \quad k, n = 0, 1, \ldots,$$

with $\boldsymbol{\varepsilon}_{-1}^{(n)} = \mathbf{0}_p$ and $\boldsymbol{\varepsilon}_0^{(n)} = \mathbf{s}_n \in \mathbb{R}^{p\times p}$, for $n = 0, 1, \ldots$

The following result was proved by Salam [35].

THEOREM 4.1. *If, for all n, $(\mathbf{s}_n)$ satisfies*

$$\boldsymbol{\alpha}_0(\mathbf{s}_n - \mathbf{s}) + \cdots + \boldsymbol{\alpha}_k(\mathbf{s}_{n+k} - \mathbf{s}) = \mathbf{0}_p,$$

or

$$(\mathbf{s}_n - \mathbf{s})\boldsymbol{\alpha}_0 + \cdots + (\mathbf{s}_{n+k} - \mathbf{s})\boldsymbol{\alpha}_k = \mathbf{0}_p,$$

where $\mathbf{s}$, the $\mathbf{s}_n$'s, and the $\boldsymbol{\alpha}_i$'s are square matrices of the same dimension and such that $\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_k$ and $\boldsymbol{\alpha}_0 + \cdots + \boldsymbol{\alpha}_k$ are nonsingular, then, for all n, $\boldsymbol{\varepsilon}_{2k}^{(n)} = \mathbf{e}_k(\mathbf{s}_n) = \mathbf{s}$.

This result does not hold true in general for rectangular matrices $\boldsymbol{\alpha}_i, \mathbf{s}_n$ and $\mathbf{s}$ when using the pseudo-inverse instead of the inverse in the matrix ε-algorithm. This is a conjecture which was discussed by Wynn in [42], and we verified numerically that it was wrong. However, the conjecture is satisfied in some special cases [18, 34]. Nevertheless, as we will see below, replacing the inverse by the pseudo-inverse can lead to interesting numerical results for square matrices (see, in particular, Example 1, Figure 1, right).

As can be seen from Theorem 4.1, only the $\boldsymbol{\varepsilon}$'s with an even lower index are related to the transformation. The other ones are intermediate computations. Thus, for saving memory, only the matrices with a lower index of the same parity can be kept and the other ones can be eliminated thus leading to the *cross-rule* first proved by Wynn in the scalar case [41]

$$[\boldsymbol{\varepsilon}_{k+2}^{(n)} - \boldsymbol{\varepsilon}_k^{(n+1)}]^{-1} + [\boldsymbol{\varepsilon}_{k-2}^{(n+2)} - \boldsymbol{\varepsilon}_k^{(n+1)}]^{-1} = [\boldsymbol{\varepsilon}_k^{(n)} - \boldsymbol{\varepsilon}_k^{(n+1)}]^{-1} + [\boldsymbol{\varepsilon}_k^{(n+2)} - \boldsymbol{\varepsilon}_k^{(n+1)}]^{-1},$$

with $\boldsymbol{\varepsilon}_{-2}^{(n)} = \boldsymbol{\infty}_p$. However, this formula is much too costly in terms of matrix inversions while the formula of the matrix ε-algorithm only requires one inversion each time.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

The $\boldsymbol{\varepsilon}$'s are displayed in a two dimensional array. The lower index k represents a column, while the upper index n designates a descending diagonal (see, for example, Table 1 in [8]). The computation of $\boldsymbol{\varepsilon}_{2k}^{(n)}$ needs the knowledge of $\mathbf{s}_0, \ldots, \mathbf{s}_{n+2k}$. Keeping all the intermediate terms of this array requires the storage of $(k+1)(2k+1)$ elements. Each term of the sequence to be transformed is computed and added one by one to the array which is then computed as far as possible. When k is fixed, adding the new element $\mathbf{s}_{n+2k+1}$, allows to compute $\boldsymbol{\varepsilon}_{2k}^{(n+1)}$ by $2k$ matrix inversions. When n is fixed and k is increased by one, two new elements $\mathbf{s}_{n+2k+1}$ and $\mathbf{s}_{n+2k+2}$ are necessary to obtain $\boldsymbol{\varepsilon}_{2k+2}^{(n)}$ which requires $2k+2$ matrix inversions. In order to save storage, the algorithm is usually implemented using a technique described, for example, in [8]. It requires to store only $2k+1$ elements plus two additional working elements.

The numerical stability of the matrix ε-algorithm is an important issue which has never been studied. This algorithm needs the inversion of the difference of two matrices whose elements can be close or be large. Thus, in both cases, the matrix to be inverted can be ill-conditioned. This is illustrated by the Example 1 below.

5. Matrix Padé approximation. Let t be a complex variable. We consider the matrix series

$$\mathbf{f}(t) = \sum_{i=0}^{\infty} \mathbf{c}_i t^i,$$

where $\mathbf{c}_i \in \mathbb{R}^{p\times s}$. A Padé approximant is the product of a polynomial with matrix coefficients (the numerator polynomial) by the inverse of another matrix polynomial (the denominator polynomial), and whose expansion in ascending powers of t agrees with that of $\mathbf{f}$ as far as possible. We will distinguish between matrix Padé-type approximants where the denominator polynomial can be arbitrarily chosen, and matrix Padé approximants where no choice is left but achieves a better order of approximation. As in the case of Shanks transformation, we will also distinguish between left and right approximants. For an extensive study of these approximants, see [13–15, 43], and also [31], [6] and [19].

5.1. Padé-type approximants. Let P_k be a polynomial of degree k with matrix coefficients $B_i \in \mathbb{R}^{s\times s}$ and Q_{k+m-1} be a matrix polynomial in $\mathbb{R}^{p\times s}$. If

$$Q_{k+m-1}(t) - \mathbf{f}(t)P_k(t) = \mathcal{O}(t^{k+m}),$$

then $Q_{k+m-1}(t)(P_k(t))^{-1}$ is called the *right matrix Padé-type approximant*, and it is denoted by $(k+m-1/k)_R$.

Now, let P_k be a polynomial of degree k with matrix coefficients $B_i \in \mathbb{R}^{p\times p}$ and Q_{k+m-1} be again a matrix polynomial in $\mathbb{R}^{p\times s}$. If

$$Q_{k+m-1}(t) - P_k(t)\mathbf{f}(t) = \mathcal{O}(t^{k+m}),$$

then $(P_k(t))^{-1}Q_{k+m-1}(t)$ is called the *left matrix Padé-type approximant*, and it is denoted by $(k+m-1/k)_L$.

Usually, left and right matrix Padé-type approximants are not identical (their dimensions are not the same if $p \neq s$) but they both use the same number of coefficients of $\mathbf{f}$, namely $\mathbf{c}_0, \ldots, \mathbf{c}_{k+m-1}$ [6].

5.2. Padé approximants. Let us now choose P_k to improve the order of approximation. We set

$$P_k(t) = B_0 + \cdots + B_{k-1}t^{k-1} + B_k t^k, \quad B_k = \mathbf{I}_s.$$

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

In the left case, if the matrices $B_i \in \mathbb{R}^{s\times s}$ are chosen so that, for some r,

$$\mathbf{c}_{k+m+i}B_0 + \cdots + \mathbf{c}_{m+i}B_k = 0, \quad i = 0, \ldots, r-1,$$

the order of approximation becomes $k+m+r$. Each of these equations is equivalent to ps scalar equations with ps^2 unknowns. For having the same number of equations and unknowns, we must have $rp = ks$. The approximants obtained are called *right matrix Padé approximants*, they are denoted by $[k+m-1/k]_R$, and they need the knowledge of $\mathbf{c}_0, \ldots, \mathbf{c}_{k+m+r-1}$, and it holds [1, pp. 429–466]

$$Q_{k+m-1}(t) - \mathbf{f}(t)P_k(t) = \mathcal{O}(t^{k+m+r}).$$

A similar treatment applies to the right case, and we have:

THEOREM 5.1. *If $p = s$ and $r = k$, the left and the right matrix Padé approximants are identical.*

Applying the matrix ε-algorithm to the partial sums of the series $\mathbf{f}$ leads to:

THEOREM 5.2. *Let $p = s$. Applying the matrix ε-algorithm to $\mathbf{s}_n = \mathbf{c}_0 + \cdots + \mathbf{c}_n t^n$ produces $\boldsymbol{\varepsilon}_{2k}^{(n)} = [n+k/k]_L = [n+k/k]_R$.*

6. Applications. Let us now show some applications of the matrix ε-algorithm to sequences of matrices. Its main drawback is the need of a matrix inversion for the computation of a new $\boldsymbol{\varepsilon}$; see Section 4. Thus, the application of the algorithm is interesting only if the gain brought by the algorithm is quite substantial.

Matrices are in bold. The matrices whose names are given in italics are taken from [22]. The dashed line represents the Euclidean norm of the error of the sequence $(\mathbf{s}_n)$ (for Example 5, the Frobenius norm was used), the plain line refers to the ε-algorithm. Errors are in log scale. In the examples where the exact solution was not mathematically known, we computed it with the corresponding MATLAB® function. The numerical results were obtained with MATLAB, version R2017a.

6.1. Example 1. The binomial iteration for computing the square root of $\mathbf{I}-\mathbf{C}$ consists of the iterations

$$\mathbf{s}_{n+1} = \frac{1}{2}(\mathbf{C} + \mathbf{s}_n^2), \quad k = 0, 1, \ldots,$$

with $\mathbf{s}_0 = 0$. The sequence $(\mathbf{s}_n)$ converges linearly to $\mathbf{s} = \mathbf{I} - (\mathbf{I}-\mathbf{C})^{1/2}$ if $\rho(\mathbf{C}) < 1$, and $(\mathbf{s}_n)$ is the sequence of the partial sums of the series

$$(\mathbf{I}-\mathbf{C})^{1/2} = \sum_{i=0}^{\infty} \binom{1/2}{i} (-\mathbf{C})^i = \mathbf{I} - \sum_{i=1}^{\infty} \gamma_i \mathbf{C}^i, \quad \gamma_i > 0,$$

up to and including the term $\mathbf{C}^n$ [21, pp. 154–157].

For $\mathbf{C}$, we took the matrix *moler* of dimension 100 divided by its spectral radius and multiplied by 0.9. The errors of the sequences $(\mathbf{s}_n)$ and $(\boldsymbol{\varepsilon}_4^{(n)})$ are plotted in Figure 1 (left) in a log scale. These results were obtained by the matrix ε-algorithm as defined by Wynn in his paper [40]. On the right of Figure 1 we show the results given by replacing, in our MATLAB code of the matrix ε-algorithm, the inverse (based on the LU decomposition) by the pseudo-inverse (which uses the SVD). We see that they are much better and that the error suddenly drops down. The spectral radius of the matrix $\mathbf{C}$ is equal to 0.9, and its condition number is 2.7796×10^{16}. After 40 iterations (that is when the maximum precision is achieved), the spectral radius of

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

$\mathbf{C}^{40}$ is 0.0148, and its condition number is equal to 3.8558×10^{19}. Inverting it by LU decomposition produces a matrix with a condition number equal to 3.7175×10^{23}, while, with the SVD it is 1.2640×10^{20}. Thus, in the ε-algorithm, computing the inverses by using the pseudo-inverses based on the SVD is more stable than LU decompositions since singular values smaller than a fixed tolerance are treated as zeros. However, it must be noticed that computing the SVD is more expensive than computing a LU decomposition. The results with the sequences $(\boldsymbol{\varepsilon}_{2k}^{(n)})$ for various values of k are quite similar.

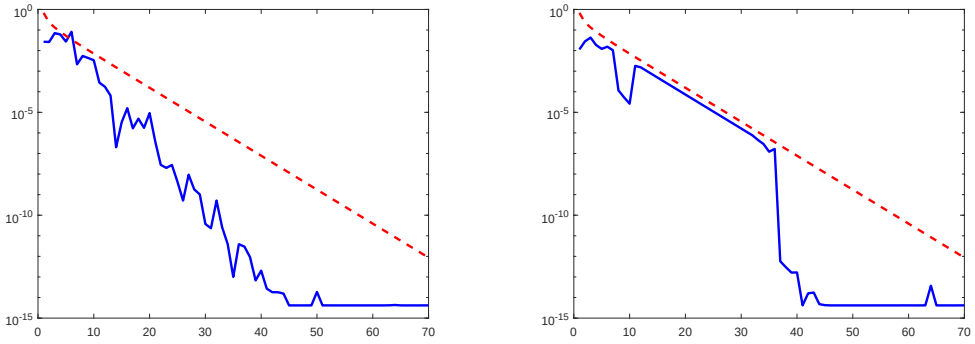

FIGURE 1. *Example 1:* $(\mathbf{s}_n)$ *(dashed line),* $(\boldsymbol{\varepsilon}_4^{(n)})$ *(plain line), with the inverse (left) and the pseudo-inverse (right).*

6.2. Example 2. We consider the matrix equation

$$f(\mathbf{s}) = \mathbf{s} + \mathbf{A}^*\mathbf{s}^{-1}\mathbf{A} - \mathbf{Q} = \mathbf{0},$$

where $\mathbf{A}, \mathbf{Q} \in \mathbb{C}^{m\times m}$ with $\mathbf{Q}$ Hermitian positive definite. We are looking for its maximal Hermitian positive definite solution $\mathbf{s}_+$, that is the matrix $\mathbf{s}_+$ such that $\mathbf{s}_+ - \mathbf{s}$ is positive semidefinite for any Hermitian solution $\mathbf{s}$ of f. For $\mathbf{Q} = \mathbf{I} + \mathbf{A}^*\mathbf{A}$, this solution is $\mathbf{s}_+ = \mathbf{I}$ if and only if $\rho(\mathbf{A}) < 1$, a result proved in [20]. Thus, an easy way to construct a numerical example is to take $\mathbf{A} = r\mathbf{S}/\rho(\mathbf{S})$ where $\mathbf{S}$ is any arbitrary matrix.

We use the following iterative method due to Guo [20]:

$$\begin{aligned} \mathbf{s}_0 &= \mathbf{Q}, \\ \mathbf{s}_{n+1} &= \mathbf{Q} - \mathbf{A}^*\mathbf{s}_n^{-1}\mathbf{A}, \quad n = 0, 1, \ldots \end{aligned}$$

This method converges slowly if the spectral radius of $\mathbf{A}$ is close to 1. For $\mathbf{S}$ we took the *prolate* matrix of dimension $m = 20$ and $r = 0.8$. The error of $(\boldsymbol{\varepsilon}_4^{(n)})$ is given in Figure 2 (left) in log scale. In this Example, we used the inverse based on the LU decomposition since it is cheaper than the SVD, and no stability problems were detected.

6.3. Example 3. We consider the computation of the exponential function $\mathbf{s} = e^{\mathbf{A}t}$. We take $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^{-1}$ so that $\mathbf{s} = \mathbf{U}e^{\mathbf{D}t}\mathbf{U}^{-1}$, where $\mathbf{D}$ is the *frank* matrix, $\mathbf{U}$ is the *orthog* matrix, both of dimension 50, and $t = -0.01$. The errors corresponding to $(\boldsymbol{\varepsilon}_6^{(n)})$ are given in Figure 2 (right). Then, the error degrades. We again used the inverse based on the LU decomposition since nothing is gained by using the pseudo-inverse.

6.4. Example 4. We consider now the partial sums $\mathbf{s}_n$ of the series $\log(\mathbf{I}+\mathbf{A}) = \mathbf{A}-\mathbf{A}^2/2+\mathbf{A}^3/3-\cdots$, where A is the *ris* matrix of dimension 100 from the matrix toolbox [22] divided by its spectral radius and

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

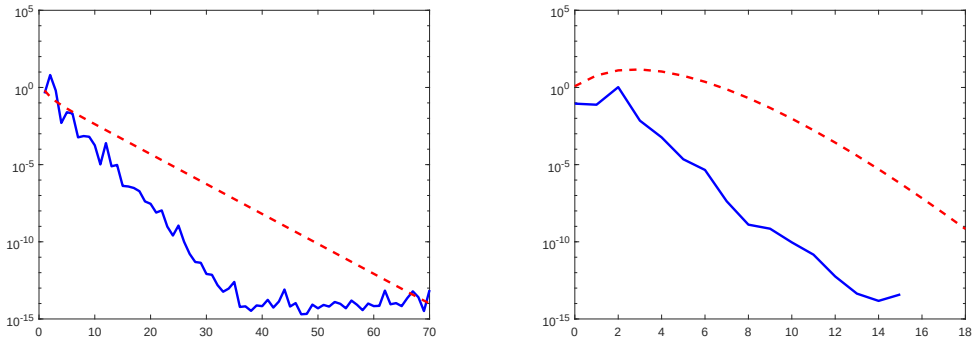

FIGURE 2. *Example 2 (left):* $(\mathbf{s}_n)$ *(dashed line),* $(\boldsymbol{\varepsilon}_4^{(n)})$ *(plain line). Example 3 (right):* $(\mathbf{s}_n)$ *(dashed line),* $(\boldsymbol{\varepsilon}_6^{(n)})$ *(plain line).*

multiplied by 0.6. The results are given by Figure 3. The dashed line represents the error, in log scale, of the partial sums of the series. The plain one corresponds to the matrix ε-algorithm. We did not show the sequence $(\boldsymbol{\varepsilon}_6^{(n)})$ which is highly oscillating and brings no improvement on $(\boldsymbol{\varepsilon}_4^{(n)})$. Similarly, the error curve of $(\boldsymbol{\varepsilon}_2^{(n)})$ almost follows that of the partial sums of the series. On Figure 3, we also see that the diagonal sequence $(\boldsymbol{\varepsilon}_{2k}^{(0)})$, which requires many matrix inversions, has no special interest. Finally, if we raise the spectral radius of A above 1, the matrix ε-algorithm diverges as the partial sums of the series, contrarily to the Simplified Topological Epsilon Algorithm which transforms a divergent series into a convergent one [8].

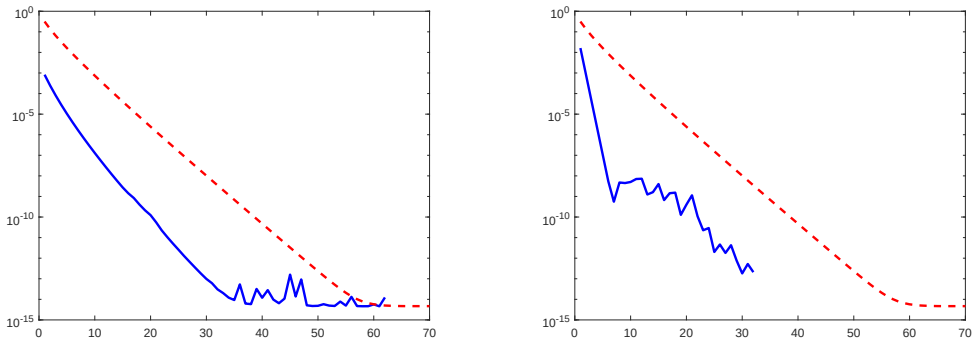

FIGURE 3. *Example 4: Left* $(\mathbf{s}_n)$ *(dashed line),* $(\boldsymbol{\varepsilon}_4^{(n)})_n$ *(plain line). Right* $(\mathbf{s}_n)$ *(dashed line),* $(\boldsymbol{\varepsilon}_{2k}^{(0)})_k$ *(plain line).*

6.5. Example 5. Let us end with a numerical example concerning rectangular matrices. We consider the Sylvester equation $\mathbf{AX}-\mathbf{XB}=\mathbf{C}$ where $\mathbf{A}\in\mathbb{R}^{p\times p}, \mathbf{B}\in\mathbb{R}^{m\times m}, \mathbf{C},\mathbf{X}\in\mathbb{R}^{p\times m}$. For solving this equation we used the Algorithm 1 given in [30]

Choose $\mathbf{X}_0\in\mathbb{R}^{np\times m}$.
Compute $\mathbf{Z}_0=\mathbf{C}+\mathbf{X}_0\mathbf{B}$
for $n=0,1,\ldots,$ until convergence
  Solve $\mathbf{AX}_{n+1}=\mathbf{Z}_n$
  $\mathbf{Z}_{n+1}=\mathbf{C}+\mathbf{X}_{n+1}\mathbf{B}$
end for

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

The sequence $(\mathbf{X}_n)$ converges q-linearly to the unique solution of the equation if $\|\mathbf{A}^{-1}\| \cdot \|\mathbf{B}\| < 1$. Thus, the convergence is quite fast, and it cannot be accelerated. Since the preceding condition is only sufficient, we took a matrix $\mathbf{B}$ such that $\|\mathbf{A}^{-1}\| \cdot \|\mathbf{B}\| = 4.5$. $\mathbf{A}$ was the *clement* matrix of dimension 10 of the matrix toolbox [22], $\mathbf{B}$ was the *condex* matrix of dimension 30. The solution $\mathbf{X}$ was taken as the *chebvand* matrix of dimension 10×30 (Figure on the left) and 10×10000 (Figure on the right), and the $\mathbf{C}$ was computed accordingly. We started from $\mathbf{X}_0 = \mathbf{0}$, and compute $\mathbf{A}^{-1}$ with `inv`. The results are given in Figure 4. The norms are the Frobenius ones. The computation of each member of the sequence $(\boldsymbol{\varepsilon}_2^{(n)})$ needs two pseudo-inverses obtained with `pinv`.

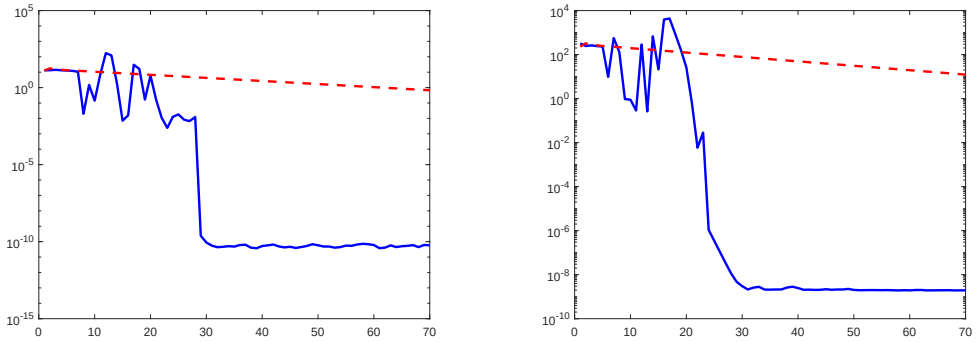

FIGURE 4. *Example 5:* $(\mathbf{X}_n)$ *(dashed line),* $(\boldsymbol{\varepsilon}_2^{(n)})$ *(plain line).*

Going to $(\boldsymbol{\varepsilon}_{2k}^{(n)})$ for $k > 1$ does not bring any further improvement, and needs the computation of two additional pseudo-inverses each time k is increased by 1.

7. Coda. In this paper, we present a complete algebraic theory of a transformation for accelerating sequences of rectangular matrices, and we relate it to Padé-type and Padé approximants for power series with matrix coefficients. This transformation is inspired by the scalar sequence transformation due to Shanks [36, 37]. In the case of square matrices, the transformation can be recursively implemented by the matrix ε-algorithm of Wynn [40]. This algorithm can be used for accelerating sequences or for solving nonlinear matrix equations. However, since it requires a matrix inversion at each step, it is really effective only for problems where it brings a sufficiently valid improvement. Despite this drawback, the theory presented in this paper is an addition to the literature on convergence acceleration and extrapolation methods.

Acknowledgment. The work of C.B. was supported in part by the Labex CEMPI (ANR-11-LABX-0007-01). The work of M.R.-Z. was partially supported by the INdAM Research group GNCS, and by the University of Padua, Project no. DOR1807909/18. The authors want to thank the reviewers for their careful reading of our paper and their remarks which helped us to clarify some explanations. We also acknowledge the efficient handling of our paper by its editor.

REFERENCES

[1] G.A. Baker Jr. and P.R. Graves-Morris. *Padé Approximants*, second edition. Cambridge University Press, Cambridge, 1996.

[2] C. Brezinski. Généralisation de la transformation de Shanks, de la table de Padé et de l'ε-algorithme. *Calcolo*, 12:317–360, 1975.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

[3] C. Brezinski. A general extrapolation algorithm. *Numer. Math.*, 35:175–187, 1980.
[4] C. Brezinski. Recursive interpolation, extrapolation and projection. *J. Comput. Appl. Math.*, 9:369–376, 1983.
[5] C. Brezinski. Other manifestations of the Schur complement. *Linear Algebra Appl.*, 111:231–247, 1988.
[6] C. Brezinski. Comparisons between vector and matrix Padé approximants. *J. Nonlinear Math. Phys.*, Supplement 2, 10:1–12, 2003.
[7] C. Brezinski and M. Redivo-Zaglia. The simplified topological ε-algorithms for accelerating sequences in a vector space. *SIAM J. Sci. Comput.*, 36:A2227–A2247, 2014.
[8] C. Brezinski and M. Redivo-Zaglia. The simplified topological ε-algorithms: software and applications. *Numer. Algorithms*, 74:1237–1260, 2017.
[9] C. Brezinski and M. Redivo-Zaglia. Extrapolation methods for the numerical solution of nonlinear Fredholm integral equations. *J. Integral Equations Appl.*, 31(1):29–57, 2019.
[10] C. Brezinski, M. Redivo-Zaglia, and Y. Saad. Shanks sequence transformations and Anderson acceleration. *SIAM Rev.*, 60:646–669, 2018.
[11] S. Cabay and L.W. Jackson. A polynomial extrapolation method for finding limits and antilimits of vector sequences. *SIAM J. Numer. Anal.*, 13:734–752, 1976.
[12] S. Cipolla, M. Redivo-Zaglia, and F. Tudisco. Extrapolation methods for fixed-point multilinear Pagerank computations. Submitted.
[13] A. Draux. *Polynômes Orthogonaux Formels. Applications*. Lecture Notes in Math., Vol. 974, Springer, Berlin, 1983.
[14] A. Draux. The Padé approximants in a non-commutative algebra and their applications. In: H. Werner and H. Bünger (editors), *Padé Approximation and its Applications, Bad Honnef, 1983*, Lecture Notes in Math., Springer, Berlin, 1071:117–131, 1984.
[15] A. Draux and B. Moalla. Rectangular matrix Padé approximants and square matrix orthogonal polynomials. *Numer. Algorithms*, 14:321–341, 1997.
[16] R.P. Eddy. Extrapolation to the limit of a vector sequence. In: P.C.C. Wang (editor), *Information Linkage between Applied Mathematics and Industry*, Academic Press, New York, 387–396, 1979.
[17] M. Fiedler. *Special Matrices and their Applications in Numerical Mathematics*. Martinus Nijhoff Publishers, Dordrecht, 1986.
[18] T.N.E. Greville. On some conjectures of P. Wynn concerning the ε-algorithm. MRC Technical Summary Report no. 877, Madison, Wisconsin, 1968.
[19] C. Gu. Matrix Padé-type approximant and directional matrix Padé approximant in the inner product space. *J. Comput. Appl. Math.*, 164/165:365–385, 2004.
[20] C.-H. Guo. Convergence rate of an iterative method for a nonlinear matrix equation. *SIAM J. Matrix Anal. Appl.*, 23:295–302, 2001.
[21] N.J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, Philadelphia, 2008.
[22] N.J. Higham. The Matrix Computation Toolbox. Available at `http://www.ma.man.ac.uk/~higham/mctoolbox`.
[23] K. Jbilou and A. Messaoudi. Matrix recursive interpolation algorithm for block linear systems: Direct methods, *Linear Algebra Appl.*, 294:137–154, 1999.
[24] K. Jbilou and A. Messaoudi. Block extrapolation methods with applications. *Appl. Numer. Math.*, 106:154–164, 2016.
[25] K. Jbilou, A. Messaoudi, and K. Tabaâ. Some Schur complement identities and applications to matrix extrapolation methods. *Linear Algebra Appl.*, 392:195–210, 2004.
[26] K. Jbilou and H. Sadok. Matrix polynomial and epsilon-type extrapolation methods with applications. *Numer. Algorithms*, 68:107–119, 2015.
[27] M. Mešina. Convergence acceleration for the iterative solution of $x = Ax + f$. *Comput. Methods Appl. Mech. Engrg.*, 10:165–173, 1977.
[28] A. Messaoudi. Matrix recursive projection and interpolation algorithms. *Linear Algebra Appl.*, 202:71–89, 1994.
[29] A. Messaoudi. Matrix extrapolation algorithms. *Linear Algebra Appl.*, 256:49–73, 1997.
[30] M. Monsalve. Block linear method for large scale Sylvester equations. *Comput. Appl. Math.*, 27:47–59, 2008.
[31] C. Pestano-Gabino and C. González-Concepción. Matrix Padé approximation of rational functions. *Numer. Algorithms*, 15:167–192, 1997.
[32] P.D. Powell. Calculating determinants of block matrices. Preprint, arXiv:1112.4379v1, 2011.
[33] B.P. Pugachëv. Acceleration of convergence of iterative processes and a method of solving systems of non-linear equations. *USSR Comput. Maths. Math. Phys.*, 17(5):199–207, 1978.
[34] L.D. Pyle. A generalized inverse ε-algorithm for constructing intersection projection matrices, with applications. *Numer. Math.*, 10:86–102, 1967.
[35] A. Salam. Non-commutative extrapolation algorithms. *Numer. Algorithms*, 7:225–251, 1994.
[36] D. Shanks. An analogy between transient and mathematical sequences and some nonlinear sequence-to-sequence transforms suggested by it. Part I. Memorandum 9994, Naval Ordnance Laboratory, White Oak, 1949.

Electronic Journal of Linear Algebra, ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 35, pp. 248-265, June 2019.

[37] D. Shanks. Non linear transformations of divergent and slowly convergent sequences. *J. Math. and Phys.*, 34:1–42, 1955.
[38] A. Sidi. Efficient implementation of minimal polynomial ans reduced rand extrapolation methods. *J. Comput. Appl. Math.*, 36:305–337, 1991.
[39] P. Wynn. On a device for computing the $e_m(S_n)$ transformation. *Math. Tables Aids Comput.*, 10:91–96, 1956.
[40] P. Wynn. Acceleration techniques for iterated vector and matrix problems. *Math. Comp.*, 16:301–322, 1962.
[41] P. Wynn. Upon systems of recursions which obtain among the quotients of the Padé table. *Numer. Math.*, 8:264–269, 1966.
[42] P. Wynn. Upon a conjecture concerning a method for solving linear equations, and certain other matters. MRC Technical Summary Report 626, Mathematics Research Center, United States Army, The University of Wisconsin, Madison, Wisconsin, 1966.
[43] G.L. Xu and A. Bultheel. Matrix Padé approximation: definitions and properties. *Linear Algebra Appl.*, 137/138:67–136, 1990.