Journal of Computational and Applied Mathematics 122 (2000) 149–165

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

Vector extrapolation methods. Applications and numerical comparison

K. Jbilou*, H. Sadok

Université du Littoral, Zone Universitaire de la Mi-voix, Batiment H. Poincaré, 50 rue F. Buisson, BP 699, F-62228 Calais Cedex, France

Received 24 November 1999; received in revised form 15 February 2000

Abstract

The present paper is a survey of the most popular vector extrapolation methods such as the reduced rank extrapolation (RRE), the minimal polynomial extrapolation (MPE), the modified minimal polynomial extrapolation (MMPE), the vector ε-algorithm (VEA) and the topological ε-algorithm (TEA). Using projectors, we derive a different interpretation of these methods and give some theoretical results. The second aim of this work is to give a numerical comparison of the vector extrapolation methods above when they are used for practical large problems such as linear and nonlinear systems of equations. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Linear systems; Nonlinear systems; Extrapolation; Projection; Vector sequences; Minimal polynomial; Epsilon-algorithm

1. Introduction

In the last decade, many iterative methods for solving large and sparse nonsymmetric linear systems of equations have been developed. The extensions of these methods to nonlinear systems have been considered. As the classical iteration processes may converge slowly, extrapolation methods are required. The aim of vector extrapolation methods is to transform a sequence of vectors generated by some process to a new one with the goal to converge faster than the initial sequence. The most popular vector extrapolation methods can be classified into two categories: the polynomial methods and the ε-algorithms. The first family contains the minimal polynomial extrapolation (MPE) method of Cabay and Jackson [8], the reduced rank extrapolation (RRE) method of Eddy [9] and Mesina [24] and the modified minimal polynomial extrapolation (MMPE) method of Sidi et al. [35], Brezinski [3] and Pugachev [25]. The second class includes the topological ε-algorithm (TEA) of

* Corresponding author.

E-mail addresses: jbilou@lma.univ-littoral.fr (K. Jbilou), sadok@lma.univ-littoral.fr (H. Sadok).

Brezinski [3] and the scalar and vector ε-algorithms (SEA and VEA) of Wynn [39,40]. Some convergence results and properties of these methods were given in [3,16,18,28,30,33–36].

Different recursive algorithms for implementing these methods were also proposed in [5,15,10,39,40]. However, in practice and for large problems, these algorithms become very unstable and are not recommended. When solving large linear and nonlinear systems, Sidi [32] gives a more stable implementation of the RRE and MPE methods using a QR decomposition while Jbilou and Sadok [19] developed an LU-implementation of the MMPE method. These techniques require low storage and work and are more stable numerically.

When applied to linearly generated vector sequences, the MPE, the RRE and the TEA methods are mathematically related to some known Krylov subspace methods. It was shown in [34] that these methods are equivalent to the method of Arnoldi [26], the generalized minimal residual method (GMRES) [27] and the method of Lanczos [21], respectively. The MMPE method is mathematically equivalent to Hessenberg method [30] and [38]. For linear problems, some numerical comparisons have been given in [11].

We note also that, when the considered sequence is not generated linearly, these extrapolation methods are still projection methods but not necessarily Krylov subspace methods [20].

An important property of the vector extrapolation methods above is that they could be applied directly to the solution of linear and nonlinear systems. This comes out from the fact that the definitions of these methods do not require an explicit knowledge of how the sequence is generated. Hence, these vector extrapolation methods are more effective for nonlinear problems [29].

For nonlinear problems, these methods do not need the use of the Jacobian of the function and have the property of quadratic convergence under some assumptions [17]. Note that for some nonlinear problems, vector extrapolation methods such as nonlinear Newton–Krylov methods fail to converge if the initial guess is "away" from a solution. In this case, some techniques such as the linear search backtracting procedure could be added to the basic algorithms; see [2].

The paper is organized as follows. In Section 2, we introduce the polynomial extrapolation methods (RRE, MPE and MMPE) by using the generalized residual. We will also see how these methods could be applied for solving linear and nonlinear systems of equations. In this case some theoretical results are given. Section 3 is devoted to the epsilon-algorithm's family (SEA, VEA and TEA). In Section 4, we give the computational steps and storage required for these methods. Some numerical experiments are given in Section 5 and a comparison with the vector extrapolation methods cited above.

In this paper, we denote by $(.,.)$ the Euclidean inner product in $\mathbb{R}^N$ and by $||.||$ the corresponding norm. For an $N \times N$ matrix A and a vector v of $\mathbb{R}^N$ the Krylov subspace $K_k(A, v)$ is the subspace generated by the vectors $v, Av, \ldots, A^{k-1}v$. I_N is the unit matrix and the Kronecker product $\otimes$ is defined by $C \otimes B = [c_{i,j}\, B]$ where B and C are two matrices.

2. The polynomial methods

2.1. Definitions of the RRE, MPE and MMPE methods

Let (s_n) be a sequence of vectors of $\mathbb{R}^N$ and consider the transformation T_k defined by

$$T_k : \mathbb{R}^N \to \mathbb{R}^N,$$
$$s_n \to t_k^{(n)}$$

with

$$t_k^{(n)} = s_n + \sum_{i=1}^{k} a_i^{(n)} g_i(n), \quad n \geqslant 0, \tag{2.1}$$

where the auxiliary vector sequences $(g_i(n))_n$; $i=1,\dots,k$, are given. The coefficients $a_i^{(n)}$ are scalars.

Let $\tilde{T}_k$ denote the new transformation obtained from T_k by

$$\tilde{t}_k^{(n)} = s_{n+1} + \sum_{i=1}^{k} a_i^{(n)} g_i(n+1), \quad n \geqslant 0. \tag{2.2}$$

For these extrapolation methods, the auxiliary sequences are such that $g_i(n) = \Delta s_{n+i-1}$, $i = 1,\dots,k$; $n \geqslant 0$, and the coefficients $a_i^{(n)}$ are the same in the two expressions (2.1) and (2.2).

We define the generalized residual of $t_k^{(n)}$ by

$$\begin{aligned}\tilde{r}(t_k^{(n)}) &= \tilde{t}_k^{(n)} - t_k^{(n)} \\ &= \Delta s_n + \sum_{i=1}^{k} a_i^{(n)} \Delta g_i(n).\end{aligned} \tag{2.3}$$

The forward difference operator Δ acts on the index n, i.e., $\Delta g_i(n) = g_i(n+1) - g_i(n)$, $i=1,\dots,k$.

We will see later that, when solving linear systems of equations, the sequence $(s_n)_n$ is generated by a linear process and then the generalized residual coincides with the classical residual.

The coefficients $a_i^{(n)}$ involved in expression (2.1) are obtained from the orthogonality relation

$$\tilde{r}(t_k^{(n)}) \perp \operatorname{span}\{y_1^{(n)},\dots,y_k^{(n)}\}, \tag{2.4}$$

where $y_i^{(n)} = \Delta s_{n+i-1}$ for the MPE; $y_i^{(n)} = \Delta^2 s_{n+i-1}$ for the RRE and $y_i^{(n)} = y_i$ for the MMPE where $y_1,\dots,y_k$ are arbitrary linearly independent vectors of $\mathbb{R}^N$.

Now, if $\tilde{W}_{k,n}$ and $\tilde{L}_{k,n}$ denote the subspaces $\tilde{W}_{k,n} = \operatorname{span}\{\Delta^2 s_n,\dots,\Delta^2 s_{n+k-1}\}$ and $\tilde{L}_{k,n} = \operatorname{span}\{y_1^{(n)},\dots,y_k^{(n)}\}$, then from (2.3) and (2.4), the generalized residuals satisfies

$$\tilde{r}(t_k^{(n)}) - \Delta s_n \in \tilde{W}_{k,n} \tag{2.5}$$

and

$$\tilde{r}(t_k^{(n)}) \perp \tilde{L}_{k,n}. \tag{2.6}$$

Conditions (2.5) and (2.6) show that the generalized residual $\tilde{r}(t_k^{(n)})$ is obtained by projecting, the vector Δs_n onto the subspace $\tilde{W}_{k,n}$, orthogonally to $\tilde{L}_{k,n}$.

In a matrix form, $\tilde{r}(t_k^{(n)})$ can be written as

$$\tilde{r}(t_k^{(n)}) = \Delta s_n - \Delta^2 S_{k,n} (L_{k,n}^{\mathrm{T}} \Delta^2 S_{k,n})^{-1} L_{k,n}^{\mathrm{T}} \Delta s_n, \tag{2.7}$$

where $L_{k,n}$, $\Delta S_{k,n}$ and $\Delta^2 S_{k,n}$ are the $k \times k$ matrices whose columns are $y_1^{(n)},\dots,y_k^{(n)}$, $\Delta s_n,\dots,\Delta s_{n+k-1}$ and $\Delta^2 s_n,\dots,\Delta^2 s_{n+k-1}$ respectively. Note that $\tilde{r}(t_k^{(n)})$ is well defined if and only if the $k \times k$ matrix

$L_{k,n}^{\mathrm{T}}\Delta^2 S_{k,n}$ is nonsingular; a necessary condition for this is that the matrices $L_{k,n}$ and $\Delta^2 S_{k,n}$ are full rank. In this case, $t_k^{(n)}$ exists and is uniquely given by

$$t_k^{(n)} = s_n - \Delta S_{k,n}(L_{k,n}^{\mathrm{T}}\Delta^2 S_{k,n})^{-1}L_{k,n}^{\mathrm{T}}\Delta s_n. \tag{2.8}$$

The approximation $t_k^{(n)}$ can also be expressed as

$$t_k^{(n)} = \sum_{j=0}^{k} \beta_j^{(n)} s_{n+j}$$

with

$$\sum_{i=0}^{k} \beta_j^{(n)} = 1$$

and

$$\sum_{j=0}^{k} \alpha_{i,j}^{(n)} \beta_j^{[n)} = 0, \quad j = 0,\ldots,k-1,$$

where the coefficients $\alpha_{i,j}^{(n)}$ are defined by

$$\alpha_{i,j}^{(n)} = (\Delta s_{n+i}, \Delta s_{n+j}) \quad \text{for the MPE method,}$$
$$\alpha_{i,j}^{(n)} = (\Delta^2 s_{n+i}, \Delta s_{n+j}) \quad \text{for the RRE method,}$$
$$\alpha_{i,j}^{(n)} = (y_{i+1}, \Delta s_{n+j}) \quad \text{for the MPE method, } i = 0,\ldots,k-1 \text{ and } j = 0,\ldots,k.$$

From these relations it is not difficult to see that $t_k^{(n)}$ can also be written as a ratio of two determinants as follows:

$$t_k^{(n)} = \begin{vmatrix} s_n & s_{n+1} & \ldots & s_{n+k} \\ \alpha_{0,0}^{(n)} & \alpha_{0,1}^{(n)} & \ldots & \alpha_{0,k}^{(n)} \\ \vdots & \vdots & & \vdots \\ \alpha_{k-1,0}^{(n)} & \alpha_{k-1,1}^{(n)} & \ldots & \alpha_{k-1,k}^{(n)} \end{vmatrix} \Bigg/ \begin{vmatrix} 1 & 1 & \ldots & 1 \\ \alpha_{0,0}^{(n)} & \alpha_{0,1}^{(n)} & \ldots & \alpha_{0,k}^{(n)} \\ \vdots & \vdots & & \vdots \\ \alpha_{k-1,0}^{(n)} & \alpha_{k-1,1}^{(n)} & \ldots & \alpha_{k-1,k}^{(n)} \end{vmatrix}. \tag{2.9}$$

The determinant in the numerator of (2.9) is the vector obtained by expanding this determinant with respect to its first row by the classical rule.

Note that the determinant in the denominator of (2.9) is equal to $\det(L_{k,n}^{\mathrm{T}}\Delta^2 S_{k,n})$ which is assumed to be nonzero. The computation of the approximation $t_k^{(n)}$ needs the values of the terms $s_n,\ s_{n+1},\ldots,s_{n+k+1}$.

2.2. Application to linear systems

Consider the system of linear equations

$$Cx = f, \tag{2.10}$$

where C is a real nonsingular $N \times N$ matrix, f is a vector of $\mathbb{R}^N$ and x^* denotes the unique solution.

Instead of applying the extrapolation methods for solving (2.10), we will use them for the preconditioned linear system

$$M^{-1}Cx = M^{-1}f, \tag{2.11}$$

where M is a nonsingular matrix.

Starting from an initial vector s_0, we construct the sequence $(s_j)_j$ by

$$s_{j+1} = Bs_j + b; \quad j = 0, 1, \ldots \tag{2.12}$$

with $B = I - A$; $A = M^{-1}C$ and $b = M^{-1}f$.

Note that if the sequence (s_j) is convergent, its limit $s = x^*$ is the solution of the linear system (2.10).

From (2.12) we have $\Delta s_j = b - As_j = r(s_j)$, the residual of the vector s_j. Therefore using (2.3) and (2.12), it follows that the generalized residual of the approximation $t_k^{(n)}$ is the true residual

$$\tilde{r}(t_k^{(n)}) = r(t_k^{(n)}) = b - At_k^{(n)}. \tag{2.13}$$

Note also that, since $\Delta^2 s_n = -A\Delta s_n$, we have $\Delta^2 S_{k,n} = -A\Delta S_{k,n}$.

For simplicity and unless specified otherwise, we set $n = 0$, we denote $t_k^{(0)} = t_k$ and we drop the index n in our notations. Let d be the degree of the minimal polynomial $\mathscr{P}_d$ of B for the vector $s_0 - x^*$ and, as $A = I - B$ is nonsingular, P_d is also the minimal polynomial of B for $r_0 = \Delta s_0$. Therefore, the matrices $\Delta S_k = [\Delta s_0, \ldots, \Delta s_{k-1}]$ and $\Delta^2 S_k = [\Delta^2 s_0, \ldots, \Delta^2 s_{k-1}]$ have full rank for $k \leqslant d$. We also note that the approximation t_d exits and is equal to the solution of the linear system (2.10).

The three extrapolation methods make use implicitly of the polynomial $\mathscr{P}_d$ and since this polynomial is not known in practice, the aim of these methods is to approximate it.

When applied to the sequence generated by (2.12), the vector extrapolation methods above produce approximations t_k such that the corresponding residuals $r_k = b - At_k$ satisfy the relations

$$r_k \in \tilde{W}_k = A\tilde{V}_k \tag{2.14}$$

and

$$r_k \perp \tilde{L}_k, \tag{2.15}$$

where $\tilde{V}_k = \text{span}\{\Delta s_0, \ldots, \Delta s_{k-1}\}$ and $\tilde{L}_k \equiv \tilde{W}_k$ for RRE, $\tilde{L}_k \equiv \tilde{V}_k$ for MPE and $\tilde{L}_k \equiv \tilde{Y}_k = \text{span}\{y_1, \ldots, y_k\}$ for MMPE where $y_1, \ldots, y_k$ are linearly independent vectors.

Note that, since $\tilde{W}_k \equiv K_k(A, Ar_0)$, the extrapolation methods above are Krylov subspace methods. RRE is an orthogonal projection and is theoretically equivalent to GMRES while MPE and MMPE are oblique projection methods and are equivalent to the method of Arnoldi and to the Hessenberg method [38], respectively. From this observation, we conclude that for $k \leqslant d$, the approximation t_k exists and is unique, unconditionally for RRE, and this is not always the case for MPE and MMPE. In fact, for the last two methods the approximation t_k $(k < d)$ exists if and only if $\det(\Delta S_k^{\mathrm{T}} \Delta^2 S_k) \neq 0$ for MPE and $\det(Y_k^{\mathrm{T}} \Delta^2 S_k) \neq 0$ for MMPE where $Y_k = [y_1, \ldots, y_k]$.

Let P_k be the orthogonal projector onto $\tilde{W}_k$. Then from (2.14) and (2.15), the residual generated by RRE can be expressed as

$$r_k^{\text{rre}} = r_0 - P_k r_0. \tag{2.16}$$

We also consider the oblique projectors Q_k and R_k onto $\tilde{W}_k$ and orthogonally to $\tilde{V}_k$ and $\tilde{Y}_k$ respectively. It follows that the residuals produced by MPE and MMPE can be written as

$$r_k^{\text{mpe}} = r_0 - Q_k r_0 \tag{2.17}$$

and

$$r_k^{\text{mmpe}} = r_0 - R_k r_0. \tag{2.18}$$

The acute angle θ_k between r_0 and the subspace $\tilde{W}_k$ is defined by

$$\cos\theta_k = \max_{z \in \tilde{W}_k - \{0\}} \left(\frac{|(r_0, z)|}{||r_0|| ||z||} \right). \tag{2.19}$$

Note that θ_k is the acute angle between the vector r_0 and $P_k r_0$.

In the sequel we give some relations satisfied by the residual norms of the three extrapolation methods.

Theorem 1. *Let ϕ_k be the acute angle between r_0 and $Q_k r_0$ and let ψ_k denote the acute angle between r_0 and $R_k r_0$. Then we have the following relations*:

(1) $||r_k^{\text{rre}}||^2 = (\sin^2\theta_k)||r_0||^2$.
(2) $||r_k^{\text{mpe}}||^2 = (\tan^2\phi_k)||r_0||^2$.
(3) $||r_k^{\text{rre}}|| \leqslant (\cos\phi_k)||r_k^{\text{mpe}}||$.

Moreover if for MMPE $y_j = r_0$ *for some* $j = 1, \ldots, k$, *then we also have*

(4) $||r_k^{\text{mmpe}}||^2 = (\tan^2\psi_k)||r_0||^2$.
(5) $||r_k^{\text{rre}}|| \leqslant (\cos\psi_k)||r_k^{\text{mmpe}}||$.

Proof. Parts (1)–(3) have been proved in [18]

(4) From (2.18), we get

$$(r_k^{\text{mmpe}}, r_k^{\text{mmpe}}) = (r_k^{\text{mmpe}}, r_0 - R_k r_0).$$

Since $(r_k^{\text{mmpe}}, r_0) = 0$, it follows that

$$\begin{aligned}(r_k^{\text{mmpe}}, r_k^{\text{mmpe}}) &= (r_k^{\text{mmpe}}, -R_k r_0)\\ &= -||r_k^{\text{mmpe}}|| ||R_k r_0|| \cos(r_k^{\text{mmpe}},\ R_k r_0)\\ &= ||r_k^{\text{mmpe}}|| ||R_k r_0|| \sin\psi_k.\end{aligned}$$

On the other hand,

$$||r_0|| = ||R_k r_0|| \cos\psi_k,$$

hence

$$||r_k^{\text{mmpe}}|| = ||r_0|| \tan\psi_k.$$

(5) Using statements (1) and (4), we get

$$\frac{||r_k^{\text{mmpe}}||^2}{||r_k^{\text{rre}}||^2} = \frac{1 - \cos^2\psi_k}{1 - \cos^2\theta_k} (\cos^2\psi_k)^{-1}.$$

But $\cos\psi_k \leqslant \cos\theta_k$, therefore

$$\|r_k^{\mathrm{rre}}\| \leqslant \|r_k^{\mathrm{mmpe}}\| \cos\psi_k. \qquad \square$$

Remark.

- From relations (1), (2) and (4) of Theorem 1, we see that the residuals of the RRE are always defined while those produced by MPE and MMPE may not exist.
- We also observe that if a stagnation occurs in RRE ($\|r_k^{\mathrm{rre}}\| = \|r_0\|$ for some $k<d$), then $\cos\theta_k = 0$ and, from (2.19), this implies that $\cos\phi_k = \cos\psi_k = 0$ and hence the approximations produced by MPE and MMPE are not defined.

When the linear process (2.12) is convergent, it is more useful in practice to apply the extrapolation methods after a fixed number p of basic iterations. We note also that, when these methods are used in their complete form, the required work and storage grow linearly with the iteration step. To overcome this drawback we use them in a cycling mode and this means that we have to restart the algorithms after a chosen number m of iterations.

The algorithm is summarized as follows:

```
1. $k = 0$, choose $x_0$ and the numbers $p$ and $m$.
2. Basic iteration
       set $t_0 = x_0$
           $z_0 = t_0$
           $z_{j+1} = Bz_j + b$, $j = 0, \ldots, p-1$.
3. Extrapolation scheme
       $s_0 = z_p$
           $s_{j+1} = Bs_j + b$, $j = 0, \ldots, m$,
           compute the approximation $t_m$ by RRE, MPE or MMPE.
4. Set $x_0 = t_m$, $k = k+1$ and go to 2.
```

Stable schemes for the computation of the approximation t_k are given in [32, 19]. In [32], Sidi gave an efficient implementation of the MPE and RRE methods which is based on the QR decomposition of the matrix ΔS_k. In [19], we used an LU decomposition of ΔS_k with a pivoting strategy. These implementations require low work and storage and are more stable numerically.

2.3. Application to nonlinear systems

Consider the system of nonlinear equations

$$G(x) = x, \tag{2.20}$$

where $G : \mathbb{R}^N \Rightarrow \mathbb{R}^N$ and let x^* be a solution of (2.20).

For any arbitrary vector x, the residual is defined by

$$r(x) = G(x) - x.$$

Let $(s_j)_j$ be the sequence of vectors generated from an initial guess s_0 as follows:

$$s_{j+1} = G(s_j), \quad j = 0, 1, \ldots . \tag{2.21}$$

Note that

$$r(s_j) = \tilde{r}(s_j) = \Delta s_j, \quad j=,1,\ldots\ .$$

As for linear problems, it is more useful to run some basic iterations before the application of an extrapolation method for solving (2.20). Note also that the storage and the evaluation of the function G increase with the iteration step k. So, in practice, it is recommended to restart the algorithms after a fixed number of iterations. Another important remark is the fact that the extrapolation methods are more efficient if they are applied to a preconditioned nonlinear system

$$\tilde{G}(x) = x, \tag{2.22}$$

where the function $\tilde{G}$ is obtained from G by some preconditioning nonlinear technique.

An extrapolation algorithm for solving the nonlinear problem (2.22) is summarized as follows:

```
1. $k = 0$, choose $x_0$ and the integers $p$ and $m$.
2. Basic iteration
      set $t_0 = x_0$
         $w_0 = t_0$
         $w_{j+1} = \tilde{G}(w_j)$, $j = 0,\ldots,p-1$.
3. Extrapolation phase
      $s_0 = w_p$;
         if $||s_1 - s_0|| < \varepsilon$ stop;
         otherwise generate $s_{j+1} = \tilde{G}(s_j)$, $j = 0,\ldots,m$,
         compute the approximation $t_m$ by RRE, MPE or MMPE;
4. set $x_0 = t_m$, $k = k+1$ and go to 2.
```

As for systems of linear equations, efficient computation of the approximation t_m produced by RRE, MPE and MMPE have been derived in [32,19]. These implementations give as an estimation of the residual norm at each iteration and it allows to stop the algorithms without having to compute the true residual which requires an extra evaluation of the function $\tilde{G}$.

Important properties of vector extrapolation methods is the fact that they do not use the knowledge of the Jacobian of the function $\tilde{G}$ and have a quadratic convergence (when they are used in their complete form).

We also note that the results of Theorem 1 are still valid for nonlinear problems by replacing in the relations of this theorem the residual r_k by the generalized residual $\tilde{r}_k$.

Vector extrapolation methods such as MMPE can also be used for computing eigenelements of a matrix [16].

3. The ε-algorithms

3.1. The scalar ε-algorithm

Let (x_n) be a scalar sequence and consider the Hankel determinant

$$H_k(x_n) = \begin{vmatrix} x_n & \dots & x_{n+k-1} \\ \vdots & \vdots & \vdots \\ x_{n+k-1} & \dots & x_{n+2k-2} \end{vmatrix}, \quad \text{with } H_0(x_n) = 0, \ \forall n.$$

Shanks's transformation [31] e_k is defined by

$$e_k(x_n) = \frac{H_{k+1}(x_n)}{H_k(\Delta^2 x_n)}. \tag{3.1}$$

For the kernel of the transformation e_k, it was proved (see [6]) that

$$\forall n, \ e_k(x_n) = x \Leftrightarrow \ \exists a_0, \dots, a_k \text{ with } a_k \neq 0 \text{ and } a_0 + \dots + a_k \neq 0 \text{ such that } \forall n,$$
$$\sum_{i=0}^{k} a_i(x_{n+i} - x) = 0.$$

To implement Shank's transformation without computing determinants, Wynn [39] discovered a simple recursion called the scalar epsilon algorithm (SEA) defined by

$$\varepsilon_{-1}^{(n)} = 0, \quad \varepsilon_0^{(n)} = x_n, \quad n = 0, 1, \dots,$$
$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + \frac{1}{\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}} \ k, \quad n = 0, 1, \dots \ .$$

The scalar ε-algorithm is related to Shanks's transformation by

$$\varepsilon_{2k}^{(n)} = e_k(x_n) \quad \text{and} \quad \varepsilon_{2k+1}^{(n)} = \frac{1}{e_k(\Delta x_n)}.$$

For more details and properties of SEA, see [6] and the references therein. For vector sequences (s_n), one can apply the scalar ε-algorithm to each component of s_n. However, one disadvantage of this technique is that it ignores the connexions between the components. Another problem is the fact that some transformed components fail to exist or may be very large numerically. These drawbacks limit the application of SEA to vector sequences.

3.2. The vector ε-algorithm

In order to generalize the scalar ε-algorithm to the vector case, we have to define the inverse of a vector. One possibility that was considered by Wynn [40] is to use the inverse defined by

$$z^{-1} = \frac{z}{||z||^2}, \quad z \in \mathbb{R}^N.$$

Therefore, for vector sequences (s_n) the vector ε-algorithm of Wynn is defined by

$$\varepsilon_{-1}^{(n)} = 0, \quad \varepsilon_0^{(n)} = s_n, \quad n = 0, 1, \dots,$$
$$\varepsilon_{k+1}^{(n)} = \varepsilon_{k-1}^{(n+1)} + [\varepsilon_k^{(n+1)} - \varepsilon_k^{(n)}]^{-1}, \quad k, n = 0, 1, \dots \ .$$

For the real case, it was proved by McLeod [23] that if $\forall n \geqslant N_0$, $\sum_{i=0}^{k} a_i(s_{n+i} - s) = 0$, with $a_k \neq 0$ and $a_0 + \dots + a_k \neq 0$, then $\varepsilon_{2k}^{(n)} = s$; $\forall n \geqslant N_0$. This result has been proved by Graves-Morris [13] in the complex case.

When applied to the vector sequence generated by (2.12), the scalar and the vector ε-algorithms give the solution of the linear system (2.10) that is $\forall n$, $\varepsilon_{2N}^{(n)} = x^*$, see [6]. As will be seen in the last section, the intermediate quantities $\varepsilon_{2k}^{(n)}$, $k < N$, are approximations of the solution x^*.

We note also that the vector ε-algorithm has been used for solving nonlinear problems by applying it to the nonlinear sequence defined by (2.21); see [7,12].

However, the vector ε-algorithm requires higher work and storage as compared to the vector polynomial methods described in Section 2. In fact, computing the approximation $\varepsilon_{2k}^{(n)}$ needs the terms $s_n, \ldots, s_{n+2k}$ which requires a storage of $2k+1$ vectors of $\mathbb{R}^N$ while the three methods (RRE, MPE and MMPE) require only $k+2$ terms $s_n, \ldots, s_{n+k+1}$. Computational work and storage requirements are given in Section 4.

3.3. The topological ε-algorithm

In [3], Brezinski proposed another generalization of the scalar ε-algorithm for vector sequences which is quite different from the vector ε-algorithm and was called the topological ε-algorithm (TEA).

This approach consists in computing approximations $e_k(s_n) = t_k^{(n)}$ of the limit or the anti-limit of the sequence (s_n) such that

$$t_k^{(n)} = s_n + \sum_{i=1}^{k} a_i^{(n)} \Delta s_{n+i-1}, \quad n \geqslant 0. \tag{3.2}$$

We consider the new transformations $\tilde{t}_{k,j}$, $j = 1, \ldots, k$ defined by

$$\tilde{t}_{k,j}^{(n)} = s_{n+j} + \sum_{i=1}^{k} a_i^{(n)} \Delta s_{n+i+j-1}, \quad j = 1, \ldots, k.$$

We set $\tilde{t}_{k,0}^{(n)} = t_k^{(n)}$ and define the jth generalized residual as follows:

$$\begin{aligned} \tilde{r}_j(t_k^{(n)}) &= \tilde{t}_{k,j}^{(n)} - \tilde{t}_{k,j-1}^{(n)} \\ &= \Delta s_{n+j-1} + \sum_{i=1}^{k} a_i^{(n)} \Delta^2 s_{n+i+j-2}, \quad j = 1, \ldots, k. \end{aligned}$$

Therefore, the coefficients involved in expression (3.2) of $t_k^{(n)}$ are computed such that each jth generalized residual is orthogonal to some chosen vector $y \in \mathbb{R}^N$, that is

$$(y, \tilde{r}_j(t_k^{(n)})) = 0, \quad j = 1, \ldots, k. \tag{3.3}$$

Hence the vector $a_n = (a_1^{(n)}, \ldots, a_k^{(n)})^{\mathrm{T}}$ is the solution of the $k \times k$ linear system (3.3) which is written as

$$T_{k,n}\, a_n = \Delta S_{k,n}^{\mathrm{T}} y, \tag{3.4}$$

where $T_{k,n}$ is the matrix whose columns are $\Delta^2 S_{k,n}^{\mathrm{T}}\, y, \ldots, \Delta^2 S_{k,n+k-1}^{\mathrm{T}}\, y$ (assumed to be nonsingular) and $\Delta^j S_{k,n}$, $j = 1,2$ are the $N \times k$ matrices whose columns are $\Delta^j s_n, \ldots, \Delta^j s_{n+k-1}$, $j = 1,2$.

Note that the $k \times k$ matrix $T_{k,n}$ is also given by the formula

$$T_{k,n} = \mathscr{S}_{k,n}\,(I_N \otimes y),$$

where $\mathscr{S}_{k,n}$ is the $k \times Nk$ matrix whose block columns are $\Delta^2 S_{k,n}^{\mathrm{T}}, \ldots, \Delta^2 S_{k,n+k-1}^{\mathrm{T}}$.

Invoking (3.2) and (3.4), $t_k^{(n)}$ can be expressed in a matrix form as

$$t_k^{(n)} = s_n - \Delta S_{k,n}\, T_{k,n}^{-1}\, \Delta S_{k,n}^{\mathrm{T}} y. \tag{3.5}$$

Using Schur's formula, $t_k^{(n)}$ can be expressed as a ratio of two determinants

$$t_k^{(n)} = \begin{vmatrix} s_n & \Delta S_{k,n} \\ \Delta S_{k,n}^{\mathrm{T}} y & T_{k,n} \end{vmatrix} \Big/ \det(T_{k,n}).$$

For the kernel of the topological ε-algorithm it is easy to see that if $\forall n$, $\exists a_0, \ldots, a_k$ with $a_k \neq 0$ and $a_0 + \cdots + a_k \neq 0$ such that $\sum_{i=0}^{k} a_i(s_{n+i} - s) = 0$, then $\forall n$, $t_k^{(n)} = s$.

The vectors $e_k(s_n) = t_k^{(n)}$ can be recursively computed by the topological ε-algorithm discovered by Brezinski [3]

$$\varepsilon_{-1}^{(n)} = 0; \quad \varepsilon_0^{(n)} = s_n, \quad n = 0, 1, \ldots,$$

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

$$\varepsilon_{2k+2}^{(n)} = \varepsilon_{2k}^{(n+1)} + \frac{\Delta\varepsilon_{2k}^{(n)}}{(\Delta\varepsilon_{2k+1}^{(n)}, \Delta\varepsilon_{2k}^{(n)})} n, \quad k = 0, 1, \ldots .$$

The forward difference operator Δ acts on the superscript n and we have

$$\varepsilon_{2k}^{(n)} = e_k(s_n) = t_k^{(n)}, \quad \text{and} \quad \varepsilon_{2k+1}^{(n)} = \frac{y}{(y, e_k(\Delta s_n))}, \quad n, k = 0, 1, \ldots .$$

We notice that, for the complex case, we can use the product $(y, z) = \sum_{i=1}^{N} y_i \bar{z}_i$, hence (y, z) is not equal to (z, y). The order of vectors in the scalar product is important, and similar methods have been studied in detail by Tan [37].

3.4. Application of VEA and TEA to linear and nonlinear systems

Consider again the system of linear equations (2.10) and let (s_n) be the sequence of vectors generated by the linear process (2.12).

Using the fact that $\Delta^2 s_{n+i} = B \Delta^2 s_{n+i-1}$, the matrix $T_{k,n}$ has now the following expression:

$$T_{k,n} = -L_k^{\mathrm{T}} A \Delta S_{k,n}, \tag{3.6}$$

where L_k is the $N \times k$ matrix whose columns are $y, B^{\mathrm{T}} y, \ldots, B^{\mathrm{T}^{k-1}} y$. As n will be a fixed integer, we set $n = 0$ for simplicity and denote $T_{k,0}$ by T_k and $\Delta S_{k,0}$ by ΔS_k.

On the other hand, it is not difficult to see that

$$\Delta S_k^{\mathrm{T}} y = L_k^{\mathrm{T}} r_0. \tag{3.7}$$

Therefore, using (3.6), (3.7) with (3.5), the kth residual produced by TEA is given by

$$r_k^{\mathrm{tea}} = r_0 - A \Delta S_k\, (L_k^{\mathrm{T}} A \Delta S_k)^{-1} L_k^{\mathrm{T}} r_0. \tag{3.8}$$

Let E_k denotes the oblique projector onto the Krylov subspace $K_k(A, Ar_0)$ and orthogonally to the Krylov subspace $K_k(B^{\mathrm{T}}, y) = K_k(A^{\mathrm{T}}, y)$. Then from (3.8) the residual generated by TEA can be written as follows:

$$r_k^{\mathrm{tea}} = r_0 - E_k r_0. \tag{3.9}$$

This shows that the topological ε-algorithm is mathematically equivalent to the method of Lanczos [4]. Note that the kth approximation defined by TEA exists if and only if the $k \times k$ matrix $L_k^{\mathrm{T}} A \Delta S_k$ is nonsingular.

The following result gives us some relations satisfied by the residual norms in the case where $y = r_0$.

Theorem 2. *Let φ_k be the acute angle between r_0 and $E_k r_0$ and let $y = r_0$. Then we have the following relations*:
(1) $\|r_k^{\text{tea}}\| = |\tan \varphi_k| \|r_0\|$; $k > 1$.
(2) $\|r_k^{\text{rre}}\| \leqslant (\cos \varphi_k) \|r_k^{\text{tea}}\|$.

Proof. (1) Follows from (3.9) and the fact that $r_0 = y$ is orthogonal to r_k^{tea}.

(2) From (2.19) we have $\cos \varphi_k \leqslant \cos \theta_k$, then using relations (1) of Theorem 1 and (1) of Theorem 2 the result follows. □

Remark.
- Relation (1) of Theorem 2 shows that the residuals of the TEA are defined if and only if $\cos \varphi_k \neq 0$.
- We also observe that, if a stagnation occurs in RRE ($\|r_k^{\text{rre}}\| = \|r_0\|$ for some k, then $\cos \theta_k = 0$ and this implies that $\cos \varphi_k = 0$, which shows that the TEA-approximation is not defined.

The topological ε-algorithm can also be applied for solving nonlinear systems of equations. For this, TEA is applied to the sequence (s_n) generated by the nonlinear process (2.22). We note that TEA does not need the knowledge of the Jacobian of the function $\tilde{G}$ and has the property of quadratic convergence [22].

When applied for the solution of linear and nonlinear problems, work and storage required by VEA and TEA grow with the iteration step. So, in practice and for large problems, the algorithms must be restarted. It is also useful to run some basic iterations before the extrapolation phase.

The application of VEA or TEA for linear and nonlinear systems leads to the following algorithm where $\tilde{G}(x)$ is to be replaced by $Bx + b$ for linear problems:

```
1. $k = 0$, choose $x_0$ and the integers $p$ and $m$.
2. Basic iteration
     set $t_0 = x_0$
       $w_0 = t_0$
       $w_{j+1} = \tilde{G}(w_j)$, $j = 0, \ldots, p-1$.
3. Extrapolation phase
     $s_0 = w_p$;
       if $\|s_1 - s_0\| < \varepsilon$ stop;
       otherwise generate $s_{j+1} = \tilde{G}(s_j)$, $j = 0, \ldots, 2m-1$,
       compute the approximation $t_m = \varepsilon_{2m}^{(0)}$ by VEA or TEA;
4. set $x_0 = t_m$, $k = k + 1$ and go to 2.
```

Table 1
Memory requirements and computational costs (multiplications and additions) for RRE, MPE, MMPE, VEA and TEA

Method	RRE	MPE	MMPE	VEA	TEA
Multiplications and additions	$2Nk^2$	$2Nk^2$	Nk^2	$10Nk^2$	$10Nk^2$
Mat–Vec with A or evaluation of $\tilde{G}$	$k+1$	$k+1$	$k+1$	$2k$	$2k$
Memory locations	$(k+1)N$	$(k+1)N$	$(k+1)N$	$(2k+1)N$	$(2k+1)N$

4. Operation count and storage

Table 1 lists the operation count (multiplications and additions) and the storage requirements to compute the approximation $t_k^{(0)}$ with RRE, MPE and MMPE and the approximation $\varepsilon_{2k}^{(0)}$ with VEA and TEA. In practice, the dimension N of vectors is very large and k is small, so we listed only the main computational effort.

For RRE and MPE, we used the QR-implementation given in [32], whereas the LU-implementation developed in [19] was used for MMPE.

To compute $t_k^{[0)}$ with the three polynomial vector extrapolation methods, the vectors $s_0, s_1, \ldots, s_{k+1}$ are required while the terms $s_0, \ldots, s_{2k}$ are needed for the computation of $\varepsilon_{2k}^{(0)}$ with VEA and TEA. So, when solving linear systems of equations, $k+1$ matrix–vector (Mat–Vec) products are required with RRE, MPE and MMPE and $2k$ matrix–vector products are needed with VEA and TEA. For nonlinear problems the comparison is still valid by replacing "Mat–Vec" with "evaluation of the function $\tilde{G}$".

All these operations are listed in Table 1.

As seen in Table 1, the vector and topological ε-algorithms are more expensive in terms of work and storage as compared to the polynomial vector extrapolation methods, namely RRE, MPE and MMPE.

The implementations given in [32,19] for RRE, MPE and MMPE allow us to compute exactly the norm of the residual at each iteration for linear systems or to estimate it for nonlinear problems without actually computing the residuals. This reduce the cost of implementation and is used to stop the algorithms when the accuracy is achieved.

5. Numerical examples

We report in this section a few numerical examples comparing the performances of RRE, MPE, MMPE, VEA and TEA. For RRE and MPE, we used the program given in [32] and for MMPE we used the implementation developed in [19]. The programs used for VEA and TEA were taken out from [6].

The tests were run in double precision on SUN Entreprise 450 SERVER using the standard F77 compiler. We have first considered one example for linear systems and one example for nonlinear systems. In these examples the starting point was chosen $x_0 = \mathrm{rand}(N,1)$ where the function rand creates an N vector with coefficients uniformly distributed in $[0,1]$.

For the TEA the vector y was also $y = \mathrm{rand}(N,1)$.

Table 2

Method	MMPE	MPE	RRE	VEA	TEA
Number of restarts	28	25	26	30	30
Residual norms	2.16d-09	2.d-09	1.d-09	9d-04	3d-01
CPU time	40	80	83	230	206

5.1. Example 1

In the first example, we derived the matrix test problem by discretizing the boundary value problem [1]

$$-u_{xx}(x,y) - u_{yy}(x,y) + 2p_1u_x(x,y) + 2p_2u_y(x,y) - p_3u(x,y) = \phi(x,y) \quad \text{on } \Omega,$$

$$u(x,y) = 1 + xy \quad \text{on } \partial\Omega,$$

by finite differences, where Ω is the unit square $\{(x,y) \in R^2,\ 0 \leqslant x, y \leqslant 1\}$ and p_1, p_2, p_3 are positive constants. The right-hand-side function $\phi(x,y)$ was chosen so that the true solution is $u(x,y) = 1 + xy$ in Ω. We used centred differences to discretize this problem on a uniform $(n+2) \times (n+2)$ grid (including grid points on the boundary). We get a matrix of size $N = n^2$.

We applied the extrapolation methods to the sequence (s_j) defined as in [14] by

$$s_{j+1} = B_\omega s_k + c_\omega, \tag{5.1}$$

where

$$c_\omega = \omega(2-\omega)(D-\omega U)^{-1}D(D-\omega L)^{-1}b, \tag{5.2}$$

$$B_\omega = (D-\omega U)^{-1}(\omega L + (1-\omega)D)(D-\omega L)^{-1}(\omega U + (1-\omega)D) \tag{5.3}$$

and $A = D - L - U$, the classical splitting decomposition.

When (s_j) converges, the fixed point of iteration (5.1) is the solution of the SSOR preconditioned system $(I - B_\omega)x = c_\omega$.

The stopping criterion was $||(I - B_\omega)x_k - c_\omega|| < 10^{-8}$ for this linear problem.

We let $n = 70$ and choose $p_1 = 1,\ p_2 = 1$ and $p_3 = 10$.

For this experiment, the system has dimension 4900×4900. The width of extrapolation is $m = 20$ and $\omega = 0.5$.

In Table 2, we give the l_2-norm of the residuals obtained at the end of each cycle and the CPU time for the five methods (MMPE, MPE, RRE, VEA and TEA).

A maximum of 30 cycles was allowed to all the algorithms. Remark that for this experiment, TEA failed to converge and for the VEA we obtained only a residual norm of $9 \cdot 10^{-4}$.

5.2. Example 2

We consider now the following nonlinear partial differential equation:

$$-u_{xx}(x,y) - u_{yy}(x,y) + 2p_1u_x(x,y) + 2p_2u_y(x,y) - p_3u(x,y) + 5e^{u(x,y)} = \phi(x,y) \quad \text{on } \Omega,$$

$$u(x,y) = 1 + xy \quad \text{on } \partial\Omega,$$

Table 3

Method	MMPE	MPE	RRE	VEA	TEA
Number of restarts	20	18	19	22	30
Residual norms	2.9d-09	9.2d-08	2.8d-08	9.6d-09	2.9d-05
CPU time	13.59	13.90	14.72	51.24	65.90

over the unit square of $\mathbb{R}^2$ with Dirichlet boundary condition. This problem is discretized by a standard five-point central difference formula with uniform grid of size $h = 1/(n+1)$. We get the following nonlinear system of dimension $N \times N$, where $N = n^2$:

$$AX + 5e^X - b = 0. \tag{5.4}$$

The right-hand-side function $\phi(x, y)$ was chosen so that the true solution is $u(x, y) = 1 + xy$ in Ω.

The sequence (s_j) is generated by using the nonlinear SSOR method. Hence we have $s_{j+1} = G(s_j)$, where

$$G(X) = B_\omega X + \omega(2-\omega)(D-\omega U)^{-1}D(D-\omega L)^{-1}(b - 5e^X),$$

the matrix B_ω is given in (5.3).

In the following tests, we compare the five extrapolation methods using the SSOR preconditioning. The stopping criterion was $||x_k - G(x_k)|| < 10^{-8}$.

In our tests, we choose $n = 72$ and hence the system has dimension $N = 4900$. With $m = 20$ and $\omega = 0.5$, we obtain the results of Table 3.

The convergence of the five extrapolation methods above is relatively sensitive to the choice of the parameter ω. We note that for this experiment, the TEA algorithm stagnates after 30 restarts. The VEA algorithm requires more CPU time as compared to the three polynomial extrapolation methods.

6. Conclusion

We have proposed a review of the most known vector extrapolation methods namely the polynomial ones (MMPE, RRE and MPE) and the ε-algorithms (TEA and VEA). We also give some numerical comparison of these methods. The numerical tests presented in this paper show the advantage of the vector polynomial methods. We note also that VEA is numerically more stable than TEA. However, the last two algorithms require more storage and operation counts as compared to the polynomial methods. The advantage of vector extrapolation methods when compared to the classical Krylov subspace methods is that they generalize in a straightforward manner from linear to nonlinear problems.

Acknowledgements

We would like to thank M. Redivo-Zaglia and C. Brezinski for providing us their programs for the ε-algorithms and A. Sidi for the RRE and MPE programs. We also wish to thank the referees for their valuable comments and suggestions.

References

[1] Z. Bai, D. Hu, L. Reichel, A Newton basis GMRES implementation, IMA J. Numer. Anal. 14 (1994) 563–581.

[2] P. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Statist. Comput. 11 (1990) 450–481.

[3] C. Brezinski, Généralisation de la transformation de Shanks, de la table de la Table de Padé et de l'epsilon-algorithm, Calcolo 12 (1975) 317–360.

[4] C. Brezinski, Padé-type Approximation and General Orthogonal Polynomials, International Series of Numerical Methods, Vol. 50, Birkhäuser, Basel, 1980.

[5] C. Brezinski, Recursive interpolation, extrapolation and projection, J. Comput. Appl. Math. 9 (1983) 369–376.

[6] C. Brezinski, M. Redivo Zaglia, Extrapolation Methods, Theory and Practice, North-Holland, Amsterdam, 1991.

[7] C. Brezinski, A.C. Rieu, The solution of systems of equations using the vector ε-algorithm and an application to boundary value problems, Math. Comp. 28 (1974) 731–741.

[8] S. Cabay, L.W. Jackson, A polynomial extrapolation method for finding limits and antilimits for vector sequences, SIAM J. Numer. Anal. 13 (1976) 734–752.

[9] 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.

[10] W.D. Ford, A. Sidi, Recursive algorithms for vector extrapolation methods, Appl. Numer. Math. 4 (6) (1988) 477–489.

[11] W. Gander, G.H. Golub, D. Gruntz, Solving linear equations by extrapolation, in: J.S. Kovalic (Ed.), Supercomputing, Nato ASI Series, Springer, Berlin, 1990.

[12] E. Gekeler, On the solution of systems of equations by the epsilon algorithm of Wynn, Math. Comp. 26 (1972) 427–436.

[13] P.R. Graves-Morris, Vector valued rational interpolants I, Numer. Math. 42 (1983) 331–348.

[14] L. Hageman, D. Young, Applied Iterative Methods, Academic Press, New York, 1981.

[15] K. Jbilou, A general projection algorithm for solving systems of linear equations, Numer. Algorithms 4 (1993) 361–377.

[16] K. Jbilou, On some vector extrapolation methods, Technical Report, ANO(305), Université de Lille1, France, 1993.

[17] K. Jbilou, H. Sadok, Some results about vector extrapolation methods and related fixed point iterations, J. Comput. Appl. Math. 36 (1991) 385–398.

[18] K. Jbilou, H. Sadok, Analysis of some vector extrapolation methods for linear systems, Numer. Math. 70 (1995) 73–89.

[19] K. Jbilou, H. Sadok, LU-implementation of the modified minimal polynomial extrapolation method, IMA J. Numer. Anal. 19 (1999) 549–561.

[20] K. Jbilou, H. Sadok, Hybrid vector sequence transformations, J. Comput. Appl. Math. 81 (1997) 257–267.

[21] C. Lanczos, Solution of systems of linear equations by minimized iterations, J. Res. Natl. Bur. Stand. 49 (1952) 33–53.

[22] H. Le Ferrand, The quadratic convergence of the topological ε-algorithm for systems of nonlinear equations, Numer. Algorithms 3 (1992) 273–284.

[23] J.B. McLeod, A note on the ε-algorithm, Computing 7 (1971) 17–24.

[24] M. Meŝina, Convergence acceleration for the iterative solution of $x = Ax + f$, Comput. Methods Appl. Mech. Eng. 10 (2) (1977) 165–173.

[25] B.P. Pugatchev, Acceleration of the convergence of iterative processes and a method for solving systems of nonlinear equations, USSR. Comput. Math. Math. Phys. 17 (1978) 199–207.

[26] Y. Saad, Krylov subspace methods for solving large unsymmetric linear systems, Math. Comp. 37 (1981) 105–126.

[27] 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.

[28] H. Sadok, Quasilinear vector extrapolation methods, Linear Algebra Appl. 190 (1993) 71–85.

[29] H. Sadok, About Henrici's transformation for accelerating vector sequences, J. Comput. Appl. Math. 29 (1990) 101–110.

[30] H. Sadok, Méthodes de projection pour les systèmes linéaires et non linéaires, Habilitation Thesis, Université de Lille1, France, 1994.

[31] D. Shanks, Nonlinear transformations of divergent and slowly convergent sequences, J. Math. Phys. 34 (1955) 1–42.
[32] A. Sidi, Efficient implementation of minimal polynomial and reduced rank extrapolation methods, J. Comput. Appl. Math. 36 (1991) 305–337.
[33] A. Sidi, Convergence and stability of minimal polynomial and reduced rank extrapolation algorithms, SIAM J. Numer. Anal. 23 (1986) 197–209.
[34] A. Sidi, Extrapolation vs. projection methods for linear systems of equations, J. Comput. Appl. Math. 22 (1) (1988) 71–88.
[35] A. Sidi, W.F. Ford, D.A. Smith, Acceleration of convergence of vector sequences, SIAM J. Numer. Anal. 23 (1986) 178–196.
[36] D.A. Smith, W.F. Ford, A. Sidi, Extrapolation methods for vector sequences, SIAM Rev. 29 (1987) 199–233; Correction, SIAM Rev. 30 (1988) 623–624.
[37] R.C.E. Tan, Implementation of the topological ε-algorithm, SIAM J. Sci. Statist. Comput. 9 (1988) 839–848.
[38] J.H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, England, 1965.
[39] P. Wynn, On a device for computing the $e_m(s_n)$ transformation, MTAC 10 (1956) 91–96.
[40] P. Wynn, Acceleration technique for iterated vector and matrix problems, Math. Comp. 16 (1962) 301–322.