Numerical Algorithms 4 (1993) 361–377

A general projection algorithm for solving systems of linear equations

Khalide Jbilou

Laboratoire d'Analyse Numérique et d'Optimisation, UFR IEEA – M3, Université des Sciences et Technologies de Lille, F-59655 Villeneuve d'Ascq Cedex, France

Communicated by P. van Dooren
Received 17 April 1992; revised 4 November 1992

In this paper, we give a general projection algorithm for implementing some known extrapolation methods such as the MPE, the RRE, the MMPE and others. We apply this algorithm to vectors generated linearly and derive new algorithms for solving systems of linear equations. We will show that these algorithms allow us to obtain known projection methods such as the Orthodir or the GCR.

Keywords: Linear systems, projection, vector sequences.

1. Introduction

In this paper, we introduce a general transformation for vector sequences covering many known extrapolation methods such as the minimal polynomial extrapolation (MPE) of Cabay and Jackson [5], the reduced rank extrapolation (RRE) of Eddy [6] and Mesina [17], the modified minimal polynomial extrapolation (MMPE) of Brezinski [1], Pugatchev [18], Sidi et al. [23] and other methods. An extensive survey of these vector extrapolation methods has been carried out by Smith et al. [24], Sidi [22], Gander et al. [11] and in a recent work by Jbilou and Sadok [16].

For computing the vectors $S_k^{(n)}$, for a fixed value of n, produced by this procedure, we give a unified algorithm that we shall call the general projection algorithm, in short the GPA. This algorithm could be used for solving systems of linear and nonlinear equations. In this paper we will be interested only in the first case.

It is known [21] that, when applied to linearly generated vector sequences, the extrapolation methods above are Krylov subspace methods and conjugate gradient type methods.

The purpose of the second part of this work is to use the GPA algorithm to obtain new algorithms for solving systems of linear equations. Using these algorithms, we will show how to derive some known projection methods such as the Orthodir of Jea and Young [25], the orthogonal residual method of Faber and Manteufel [9] and the generalized conjugate residual method of Eisenstat et al. [7]. This last result could be considered as an extension of a result given by Sidi [21] who showed that the RRE and the MPE are mathematically equivalent to the GCR and the method of Arnoldi described by Saad in [19].

In the last section, we apply the Sβ algorithm [15] for solving linear systems and give the connection with the Orthodir method.

Finally, let us mention that, when applied to vectors generated linearly, all the extrapolation methods above terminate in a finite number of iterations (see [21] and [11]).

2. Description of the algorithm

Let $\{s_n\}, \{g_i(n)\}, (i = 1, \ldots, k)$ be sequences of $\mathbb{R}^p$ (or $\mathbb{C}^p$), and $\{y_{m+1}, \ldots, y_{m+k}\}$ a set of linearly independent vectors of $\mathbb{R}^p$ where m is a fixed integer. We denote $\langle .,. \rangle$ the usual inner product in $\mathbb{R}^p$.

We consider the transformation ${}_mS_k : (s_n) \rightarrow ({}_mS_k^{(n)})$, defined by

$$ {}_mS_k^{(n)} - s_n = \sum_{i=1}^{k} a_i g_i(n) \,, \tag{2.1}$$

where the a_i's are such that

$$ \Delta s_n + \sum_{i=1}^{k} a_i \Delta g_i(n) \in span\{y_{m+1}, \ldots, y_{m+k}\}^{\perp} \,. \tag{2.2}$$

As we will consider only the case $m = 0$ which corresponds to the MMPE and $m = n$ for the RRE and the MPE, we will drop the index m and set ${}_mS_k^{(n)} = S_k^{(n)}$.

Let $\tilde{\mathbf{V}}_k$ and $\tilde{\mathbf{Y}}_k$ be the subspaces of $\mathbb{R}^p$ defined by $\tilde{\mathbf{V}}_k = span\{g_1(n), \ldots, g_k(n)\}$ and $\tilde{\mathbf{Y}}_k = span\{y_{m+1}, \ldots, y_{m+k}\}$. $\mathbf{V}_k$ and $\mathbf{Y}_k$ denote the matrices whose columns are respectively $g_1(n), \ldots, g_k(n)$ and $y_{m+1}, \ldots, y_{m+k}$, and we assume that $dim(\tilde{\mathbf{V}}_k) = dim(\tilde{\mathbf{Y}}_k) = k \leqslant p$. Now, if we set

$$ \alpha_k^{(n)} = \Delta s_n + \sum_{i=1}^{k} a_i \Delta g_i(n) \,, \tag{2.3}$$

then (2.1) and (2.2) can be written as

$$ S_k^{(n)} - s_n \in \tilde{\mathbf{V}}_k \,, \tag{2.4}$$

$$ \alpha_k^{(n)} \in \tilde{\mathbf{Y}}_k^{\perp} \,, \tag{2.5}$$

which gives us the following system of k equations with the k unknowns $a_1, \ldots, a_k$:

$$\langle y_{m+j}, \Delta s_n \rangle = -\sum_{i=1}^{k} a_i \langle y_{m+j}, \Delta g_i(n) \rangle, \quad j = 1, \ldots, k.$$

This system can be expressed in matrix form as

$$\begin{pmatrix} \langle j_{m+1}, \Delta g_1(n) \rangle & \ldots & \langle y_{m+1}, \Delta g_k(n) \rangle \\ \vdots & \vdots & \vdots \\ \langle y_{m+k}, \Delta g_1(n) \rangle & \ldots & \langle y_{m+k}, \Delta g_k(n) \rangle \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix} = - \begin{pmatrix} \langle y_{m+1}, \Delta s_n \rangle \\ \vdots \\ \langle y_{m+k}, \Delta s_n \rangle \end{pmatrix}.$$

If the determinant of this system is different from zero (which will be assumed in the sequel) then its solution is given by

$$a = -(\mathbf{Y}_k^{\mathrm{T}} \mathbf{W}_k)^{-1} \mathbf{Y}_k^{\mathrm{T}} \Delta s_n, \tag{2.6}$$

where $\mathbf{W}_k$ is the matrix whose columns are $\Delta g_1(n), \ldots, \Delta g_k(n)$. But,

$$S_k^{(n)} = s_n + \sum_{i=1}^{k} a_i g_i(n), \tag{2.7}$$

then

$$S_k^{(n)} = s_n - \mathbf{V}_k (\mathbf{Y}_k^{\mathrm{T}} \mathbf{W}_k)^{-1} \mathbf{Y}_k^{\mathrm{T}} \Delta s_n. \tag{2.8}$$

Now, using Schur's formula [3], $S_k^{(n)}$ can be expressed as a ratio of two determinants

$$S_k^{(n)} = \frac{\begin{vmatrix} s_n & g_1(n) & \ldots & g_k(n) \\ \langle y_{m+1}, \Delta s_n \rangle & \langle y_{m+1}, \Delta g_1(n) \rangle & \ldots & \langle y_{m+1}, \Delta g_k(n) \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle y_{m+k}, \Delta s_n \rangle & \langle y_{m+k}, \Delta g_1(n) \rangle & \ldots & \langle y_{m+k}, \Delta g_k(n) \rangle \end{vmatrix}}{\begin{vmatrix} \langle y_{m+1}, \Delta g_1(n) \rangle & \ldots & \langle y_{m+1}, \Delta g_k(n) \rangle \\ \vdots & \vdots & \vdots \\ \langle y_{m+k}, \Delta g_1(n) \rangle & \ldots & \langle y_{m+k}, \Delta g_k(n) \rangle \end{vmatrix}}, \tag{2.9}$$

where the determinant in the numerator is a vector obtained by expanding it with respect to its first row.

This transformation covers many vector sequence transformations and projection methods:

(a) if $m = n, g_i(n) = \Delta s_{n+i-1}$ and $y_{n+j} = \Delta s_{n+j-1}$, we obtain the MPE (Minimal Polynomial Extrapolation method) of Cabay and Jackson [5];
(b) if $m =, g_i(n) = \Delta s_{n+i-1}$ and $y_{n+j} = \Delta^2 s_{n+j-1}$, then $S_k^{(n)}$ are the vectors given by the RRE (Reduced Rank Extrapolation method) of Eddy [6] and Mesina [17];
(c) if $m = n, g_i(n) = \Delta^2 s_{n+i-1}$ and $y_{n+j} = \Delta^q S_{n+j-1}$, where q is an integer, we have a transformation given by Germain-Bonne [12];
(d) if $m = 0, g_i(n) = \Delta s_{n+i-1}$ and $\{y_1, \ldots, y_k\}$ a set of linearly independent vectors, we obtain the MMPE (Modified Minimal Polynomial Extrapolation method) of Sidi et al. [23], Pugatchev [18] and Brezinski [1];
(e) if $s_n = t_1^{(n)}$ and $g_i(n) = t_2^{(n+i-1)} - t_1^{(n+i-1)}$, where t_1 and t_2 are two vector sequence transformations, then $S_k^{(n)}$ is identical to the vector composite transformation given in [20] and [4].

Let us first notice that for the MMPE, we can use the H-algorithm [4] or the Sβ algorithm [14]. The MPE and RRE can also be implemented by some algorithms given by Ford and Sidi [10]. Our aim here is to derive a unified algorithm for computing recursively the $S_k^{(n)}$'s for a fixed value of n and increasing k. It will be used to derive new algorithms for solving systems of linear equations.

The algorithm that will be given in this section is obtained from the RPA of Brezinski [2].

Let

$$y = \begin{pmatrix} s_n \\ s_{n+1} \end{pmatrix}, \quad x_i = \begin{pmatrix} g_i(n) \\ g_i(n+1) \end{pmatrix} \quad \text{for } i = 1, \ldots, k,$$

and

$$z_j = \begin{pmatrix} -y_j \\ y_j \end{pmatrix}, \quad j = m+1, \ldots, m+k.$$

For any vector $v = (v_1, v_2)^{\mathrm{T}}$ of $\mathbb{R}^{2p}$ where $v_1, v_2 \in \mathbb{R}^p$, we denote $(.,.)$ the Euclidian inner product in $\mathbb{R}^{2p}$ and then

$$(z_j, v) = \langle y_j, v_2 - v_1 \rangle .$$

Let us consider the vectors E_k of $\mathbb{R}^{2p}$ defined by the ratio of determinants

$$E_k = \frac{\begin{vmatrix} y & x_1 & \cdots & x_k \\ (z_{m+1}, y) & (z_{m+1}, x_1) & \cdots & (z_{m+1}, x_k) \\ \vdots & \vdots & & \vdots \\ (z_{m+k}, y) & (z_{m+k}, x_1) & \cdots & (z_{m+k}, x_k) \end{vmatrix}}{\begin{vmatrix} (z_{m+1}, x_1) & \cdots & (z_{m+1}, x_k) \\ \vdots & & \vdots \\ (z_{m+k}, x_1) & \cdots & (z_{m+k}, x_k) \end{vmatrix}}. \tag{2.10}$$

Let $f_{k,i}$, $i>k$ be the vector of oR^{2p} whose determinantal representation is obtained from that of E_k by replacing, in the numerator of (2.10), the first column by $[x_i, (z_{m+1}, x_i), \ldots, (z_{m+k}, x_i)]^{\mathrm{T}}$. Using the RPA, E_k can be recursively computed by

$$\begin{cases} E_0 = y, \quad f_{0,i} = x_i, & i \geqslant 1, \\ E_k = E_{k-1} - \dfrac{(z_{m+k}, E_{k-1})}{(z_{m+k}, f_{k-1,k})} f_{k-1,k}, & \\ f_{k,i} = f_{k-1,i} - \dfrac{(z_{m+k}, f_{k_1,i})}{(z_{m+k}, f_{k-1,k})} f_{k-1,k}, & i = k+1, \ldots . \end{cases} \tag{2.11}$$

Now for $j = 1, \ldots, k$ we have

$$(z_{m+j}, y) = \langle y_{m+j}, \Delta s_n \rangle \quad \text{and} \quad (z_{m+j}, x_i) = \langle y_{m+j}, \Delta g_i(n) \rangle . \tag{2.12}$$

Using (2.12) in the expression (2.10) and the fact that the first row in the determinant of the numerator of (2.10) is formed by the vectors $y = (s_n, s_{n+1})^{\mathrm{T}}$, $x_i = (g_i(n), g_i(n+1))^{\mathrm{T}}$ for $i = 1, \ldots, k$, we can write $E_k(S_k^{(n)}, \tilde{S}_k^{(n)})^{\mathrm{T}}$, where $\tilde{S}_k^{(n)}$ is obtained from $S_k^{(n)}$ by replacing the first row in the determinant of the numerator of (2.9) by $s_{n+1}, g_1(n+1), \ldots, g_k(n+1)$.
Similarly, we have $f_{k,i} = (g_{k,i}, \tilde{g}_{k,i})^{\mathrm{T}}$, where $g_{k,i}$ (respectively $\tilde{g}_{k,i}$) is the vector of $\mathbb{R}^p$ corresponding to the first (respectively second) p components of $f_{k,i}$. Therefore $g_{k,i}$ can be expressed as

$$g_{k,i} = \frac{1}{D_k} \begin{vmatrix} g_i(n) & g_1(n) & \cdots & g_k(n) \\ \langle y_{m+1}, \Delta g_i(n) \rangle & \langle y_{m+1}, \Delta g_1(n) \rangle & \cdots & \langle y_{m+1}, \Delta g_k(n) \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle y_{m+k}, \Delta g_i(n) \rangle & \langle y_{m+k}, \Delta g_1(n) \rangle & \cdots & \langle y_{m+k}, \Delta g_k(n) \rangle \end{vmatrix}, \tag{2.13}$$

where D_k is the determinant obtained from the preceding one by eliminating the first row and the first column. On the other hand, from (2.3) and (2.6), $\alpha_k^{(n)}$ can also be expressed as

$$\alpha_k^{(n)} = \frac{1}{D_k} \begin{vmatrix} \Delta s_n & g_1(n) & \cdots & g_k(n) \\ \langle y_{m+1}, \Delta s_n \rangle & \langle y_{m+1}, \Delta g_1(n) \rangle & \cdots & \langle y_{m+1}, \Delta g_k(n) \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle y_{m+k}, \Delta s_n \rangle & \langle y_{m+k}, \Delta g_1(n) \rangle & \cdots & \langle y_{m+k}, \Delta g_k(n) \rangle \end{vmatrix}. \tag{2.14}$$

Applying again the RPA, $\alpha_k^{(n)}$ is recursively computed by

$$\begin{cases} \alpha_0^{(n)} = \Delta s_n, \quad h_{0,i} = \Delta g_i(n), & i \geqslant 1, \\ \alpha_k^{(n)} = \alpha_{k-1}^{(n)} - \dfrac{\langle y_{m+k}, \alpha_{k-1}^{(n)} \rangle}{\langle y_{m+k}, h_{k-1,k} \rangle} h_{k-1,k}, & \\ h_{k,i} = h_{k-1,i} - \frac{\langle y_{m+k}, h_{k-1,i} \rangle}{\langle y_{m+k}, h_{k-1,k} \rangle} h_{k-1,k}, & i = k+1, \ldots, \end{cases} \tag{2.15}$$

where $h_{k,i}$ is the vector of $\mathbb{R}^p$ obtained from $\alpha_k^{(n)}$ by replacing the first column of the numerator in (2.14) by $(\Delta g_i(n), \langle y_{m+1}, \Delta g_i(n) \rangle, \ldots, \langle y_{m+k}, \Delta g_i(n) \rangle)^{\mathrm{T}}$.

Thus from the definitions of $g_{k,i}$ and $\tilde{g}_{k,i}$, it is easy to see that

$$h_{k,i} = \tilde{g}_{k,i} - g_{k,i}, \tag{2.16}$$

and for $j = m+1, \ldots, m+k$

$$\langle y_j, h_{k,i} \rangle = 0. \tag{2.17}$$

From (2.9) and (2.14), we also have

$$\alpha_k^{(n)} = \tilde{S}_k^{(n)} - S_k^{(n)}, \tag{2.18}$$

and for $j = m+1, \ldots, m+k$

$$\langle y_j, \alpha_k^{(n)} \rangle = 0. \tag{2.19}$$

If we use (2.16) and (2.18) in the algorithm (2.11) and if we consider only the first p components of the vectors E_k and $f_{k,i}$, the $S_k^{(n)}$'s can be computed by

$$\begin{cases} s_0^{(n)} = s_n, \quad \alpha_0^{(n)} = \Delta s_n, \quad g_{0,i} = g_i(n), \quad h_{0,i} = \Delta g_i(n), & i \geqslant 1, \\ S_k^{(n)} = S_{k-1}^{(n)} - a_k^{(n)} g_{k-1,k}, & \\ \alpha_k^{(n)} = \alpha_{k-1}^{(n)} - a_k^{(n)} h_{k-1,k}, & \\ h_{k,i} = h_{k-1,i} - d_k^{(i)} h_{k-1,k}, & i = k+1, \ldots, \\ g_{k,i} = g_{k-1,i} - d_k^{(i)} g_{k-1,k}, & i = k+1, \ldots, \end{cases} \tag{2.20}$$

where

$$a_k^{(n)} = \frac{\langle y_{m+k}, \alpha_{k-1}^{(n)} \rangle}{\langle y_{m+k}, h_{k-1,k} \rangle} \quad \text{and} \quad d_k^{(i)} = \frac{\langle y_{m+k}, h_{k-1,i} \rangle}{\langle y_{m+k}, h_{k-1,k} \rangle}.$$

This algorithm will be called the "General Projection Algorithm" (GPA in short).

In the next section, we will apply the MMPE and RRE to vectors generated by a matrix iterative method. Using the GPA, we obtain new algorithms for solving systems of linear equations. We shall show that they allow us to derive some known projection algorithms such as the Orthodir of Jea and Young [25], the Orthogonal Residual method of Faber and Manteufel [9] and the GCR [7].

We notice that the GPA could be used for solving systems of linear and non-linear equations. In this paper, we restrict ourselves to the first case.

3. Application to linear systems

3.1. DEFINITIONS AND NOTATIONS

Consider the system of linear equations

$$Ax = f, \tag{3.1}$$

where A is a nonsingular matrix of $\mathbb{R}^{p\times p}$ and f a vector of $\mathbb{R}^p$. If we set

$$A = M - N, \tag{3.2}$$

where M is a nonsingular matrix, then (3.1) can be written as

$$x = M^{-1}Nx + M^{-1}f. \tag{3.3}$$

For a given vector s_0, we generate the sequence $s_1, s_2, \ldots$ by

$$s_{j+1} = Bs_j + b, \tag{3.4}$$

with $B = M^{-1}N$ and $b = M^{-1}f$. We define the residual $r(x)$ for a vector x by

$$r(x) = b - Cx, \tag{3.5}$$

where $C = I - B = M^{-1}A$. Let us notice that the system (3.1) is equivalent to the system

$$Cx = b. \tag{3.6}$$

From (3.4) and (3.5) it is easy to see that

$$\Delta s_{j+1} = B\Delta s_j = B^{j+1}\Delta s_0, \quad \Delta^2 s_j = -C\Delta s_j, \tag{3.7}$$

and

$$r(s_j) = b - Cs_j = \Delta s_j, \quad j = 0, 1, \ldots. \tag{3.8}$$

In what follows, we will apply the MMPE and the RRE to the sequence generated by (3.4) for solving the system (3.6) and show how to simplify the GPA in this case.

3.2. THE GPA-MMPE

We shall now consider the case $m = 0$ and $g_i(n) = \Delta s_{n+i-1}$ for $i = 1, \ldots, k$. If we set $S_k^{(0)} = x_k$ and $s_0 = x_0$, then from (2.9) we have

$$x_k = \frac{1}{det(\mathbf{Y}_k^{\mathrm{T}} \Delta^2 S_k)} \begin{vmatrix} s_0 & \Delta s_0 & \dots & \Delta s_{k-1} \\ \langle y_1, \Delta s_0 \rangle & \langle y_1, \Delta^2 s_0 \rangle & \dots & \langle y_1, \Delta^2 s_{k-1} \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle y_k, \Delta s_0 \rangle & \langle y_k, \Delta^2 s_0 \rangle & \dots & \langle y_k, \Delta^2 s_{k-1} \rangle \end{vmatrix}, \tag{3.9}$$

where $\Delta^2 S_k$ is the $p \times k$ matrix whose columns are $\Delta^2 s_0, \dots, \Delta^2 s_{k-1}$. Using (3.7), it follows that $x_k - x_0 \in span\{r_0, Br_0, \dots, B^{k-1} r_0\}$. From (2.14), $\alpha_k^{(0)}$ can also be expressed as

$$\alpha_k^{(0)} = \frac{1}{det(\mathbf{Y}_k^{\mathrm{T}} \Delta^2 S_k)} \begin{vmatrix} \Delta s_0 & \Delta^2 s_0 & \dots & \Delta^2 s_{k-1} \\ \langle y_1, \Delta s_0 \rangle & \langle y_1, \Delta^2 s_0 \rangle & \dots & \langle y_1, \Delta^2 s_{k-1} \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle y_k, \Delta s_0 \rangle & \langle y_k, \Delta^2 s_0 \rangle & \dots & \langle y_k, \Delta^2 s_{k-1} \rangle \end{vmatrix}. \tag{3.10}$$

Note that x_k and $\alpha_k^{(0)}$ exist if and only if the matrix $\mathbf{Y}_k^{\mathrm{T}} \Delta^2 S_k$ is nonsingular.

Now, since $\Delta^2 s_j = -C \Delta s_j$ and $\Delta s_0 = b - C s_0$, we obtain from (3.9) and (3.10) the relation

$$\alpha_k^{(0)} = b - C x_k = r_k . \tag{3.11}$$

On the other hand, using (3.7) in the determinantal expression of $h_{i,k}$, we have for $k = 0, 1, \dots$ and $i = 1, \dots, k-1$ the relation

$$h_{i,k} = -C g_{i,k} . \tag{3.12}$$

Using (3.12) in the GPA, we obtain the following new algorithm

$$\begin{aligned} g_{0,i} &= \Delta s_i, \quad h_{0,i} = \Delta^2 s_i, \\ x_k &= x_{k-1} - \lambda_k g_{k-1,k}, \\ r_k &= r_{k-1} - \lambda_k h_{k-1,k}, \\ g_{i,k+1} &= g_{i-1,k+1} - \lambda_i^{(k)} g_{i-1,i}, \\ h_{i,k+1} &= h_{i-1,k+1} - \lambda_i^{(k)} h_{i-1,i}, \end{aligned}$$

where

$$\lambda_k = \frac{\langle y_k, r_{k-1} \rangle}{\langle y_k, h_{k-1,k} \rangle}, \tag{3.13}$$

and for $i = 1, \dots, k$

$$\lambda_i^{(k)} = \frac{\langle y_i, h_{i-1,k+1} \rangle}{\langle y_i, h_{i-1,i} \rangle}. \tag{3.14}$$

This algorithm will be called the GPA-MMPE. Let us notice that $(y_k, h_{k-1,k}) = (-1)^{k-1} det(\mathbf{Y}_k^{\mathrm{T}} \Delta^2 S_k)$. Thus if, for some k, the matrix $\mathbf{Y}_k^{\mathrm{T}} \Delta^2 S_k$ is singular, a breakdown occurs in the algorithm. It is not our purpose in this paper to develop this case.

Note that since $h_{i,k} = -C g_{i,k}$ for $i < k$, the GPA-MMPE algorithm can be written by using only the $g_{i,k}$'s. Although it is a compact form, this way needs more operations. Computing x_k by the GPA-MMPE requires $(3k+4)p + pZ$ multiplications per step. pZ denotes the number of nonzero elements in the matrix C. If we take $y_j = e_j$, where $\{e_j\}, j = 1, \ldots, p$, is the canonical basis of $\mathbb{R}^p$, the number of multiplications could be reduced to $(2k+2)p + pZ$ per step.

Next, we shall show that the Orthodir method could be derived from the GPA-MMPE. Let us first establish the following result.

THEOREM 1

Let d be the degree of the minimal polynomial of B for the vector Δs_0. Let $z_k = (-1)^k g_{k,k+1}$, then $\forall k \leqslant d$

(a) $\langle y_j, C z_k \rangle = 0, \quad j = 1, \ldots, k$;

(b) $\langle y_k, C z_{k-1} \rangle \neq 0$;

(c) $span\{z_0, \ldots, z_{k-1}\} = span\{\Delta s_0, \ldots, \Delta s_{k-1}\}$;

(d) $z_k - C z_{k-1} \in span\{z_0, \ldots, z_{k-1}\}$.

Proof

(a) follows directly from (2.17) and (3.12).

(b) We noticed that $\langle y_k, h_{k-1,k} \rangle = (-1)^{k-1} det(\mathbf{Y}_k^{\mathrm{T}} \Delta^2 S_k)$. Now, since $h_{k-1,k} = -C g_{k-1,k} = (-1)^k C z_{k-1}$, we have $\langle y_k, C z_{k-1} \rangle = -det(\mathbf{Y}_k^{\mathrm{T}} \Delta^2 S_k)$ which was assumed to be different from 0.

(c) We have $z_i = (-1)^i g_{i,i+1}$ which can be developed as

$$z_i = (-1)^i \Delta s_i + \sum_{j=0}^{i-1} (-1)^i b_j^{(i)} \Delta s_j . \tag{3.15}$$

Therefore $z_i \in span\{\Delta s_0, \ldots, \Delta s_{k-1}\}$ for $i = 0, \ldots, k-1$. We shall show now that $z_0, \ldots, z_{k-1}$ are linearly independent.

Let $a_0 z_0 + a_1 z_1 + \ldots a_{k-1} z_{k-1} = 0$. Multiplying on the left by the matrix C and by the vector y_j, we get

$$a_0 \langle y_j, C z_0 \rangle + a_1 \langle y_j, C z_1 \rangle + \ldots + a_{k-1} \langle y_j, C z_{k-1} \rangle = 0, \quad j = 1, \ldots, k. \tag{3.16}$$

But since $\langle y_j, C z_i \rangle = 0$ for $j \leqslant i$ and $\langle y_j, C z_{j-1} \rangle \neq 0$, it follows that $a_0 = a_1 = \ldots = a_{k-1} = 0$, thus $span\{z_0, \ldots, z_{k-1}\} = span\{\Delta s_0, \ldots, \Delta s_{k-1}\}$.

(d) Coming back to the definition of z_k we have

$$z_k = (-1)^k g_{k,k+1} = (-1)^K \Delta S_K + (-1)^k \sum_{i=0}^{k-1} b_i^{(k)} \Delta s_i \tag{3.17}$$

and

$$Cz_{k-1} = (-1)^{k-1} C\Delta s_{k-1} + (-1)^{k-1} \sum_{i=0}^{k-2} b_i^{(k-1)} C\Delta s_i \,. \tag{3.18}$$

But

$$C\Delta s_{k-1} = \Delta s_{k-1} - \Delta s_k$$

and therefore

$$Cz_{k-1} = (-1)^k \Delta s_k + (-1)^{k-1} \Delta s_{k-1} + (-1)^{k-1} \sum_{i=0}^{k-2} b_i^{(k-1)} (\Delta s_i - \Delta s_{i+1}) \,. \tag{3.19}$$

Subtracting (3.19) from (3.17), it follows that

$$z_k - Cz_{k-1} = \sum_{i=0}^{k-1} c_i^{(k)} \Delta s_i \,, \tag{3.20}$$

which shows that $z_k - Cz_{k-1} \in span\{\Delta s_0, \ldots, \Delta s_{k-1}\} = span\{z_0, \ldots, z_{k-1}\}$. □

Expressing $z_k - Cz_{k-1}$ in the basis $\{z_0, \ldots, z_{k-1}\}$, we obtain

$$z_k = Cz_{k-1} + \sum_{i=0}^{k-1} \gamma_i^{(k)} z_i \,. \tag{3.21}$$

If we multiply on the left by the matrix C and again by the vector y_j, we get the following linear system of equations

$$\langle y_j, Cz_k \rangle = \langle y_j, C^2 z_{k-1} \rangle + \sum_{i=0}^{k-1} \gamma_i^{(k)} \langle y_j, Cz_i \rangle, \quad j = 1, \ldots, k \,. \tag{3.22}$$

Thanks to (a) and (b) of theorem 1, (3.22) is a lower triangular system of linear equations with the unknowns $\gamma_0^{(k)}, \ldots, \gamma_{k-1}^{(k)}$ given by

$$\gamma_0^{(k)} = -\frac{\langle y_1, C^2 z_{k-1} \rangle}{\langle y_1, Cz_0 \rangle} \,, \tag{3.23}$$

and for $i = 1, \ldots, k-1$

$$\gamma_i^{(k)} = \frac{-\langle y_{i+1}, C^2 z_{k-1} \rangle - \sum_{j=0}^{i-1} \gamma_j^{(k)} \langle y_{i+1}, Cz_j \rangle}{\langle y_{i+1}, Cz_i \rangle} \,. \tag{3.24}$$

Finally the algorithm derived from the GPA-MMPE is as follows

$$\begin{cases} x_k = x_{k-1} + \lambda'_k z_{k-1}\,, \\ r_k = r_{k-1} - \lambda'_k C z_{k-1}\,, \\ z_k = C z_{k-1} + \sum_{i=0}^{k-1} \gamma_i^{(k)} z_i\,, \end{cases} \tag{3.25}$$

where

$$\lambda'_k = \frac{\langle y_k, r_{k-1} \rangle}{\langle y_k, z_{k-1} \rangle}\,,$$

and the $\gamma_i^{(k)}$'s are defined by (3.23) and (3.24).

If we set $y_{i+1} = Z^{\mathrm{T}} z_i$ for $i = 0, \ldots, k-1$, where Z is an auxiliary matrix such that ZC is positive real, i.e. its symmetric part is positive definite, (3.25) is the Orthodir algorithm described by Young and Jea in [25].

In the previous algorithm, let us replace the usual inner product by the new one defined by $\langle x, y \rangle_H = \langle x, Hy \rangle$ where x and y are two vectors of $\mathbb{R}^p$ and H is a symmetric positive definite matrix. If we take $y_{i+1} = z_i$ for $i = 0, \ldots, k-1$, then the algorithm (3.25) reduces to the orthogonal residual method described by Faber and Manteufel in [9] and Elman [8].

This shows that the MMPE is mathematically equivalent to the Orthodir and to the orthogonal residual method.

3.3. THE GAP-RRE

In order to obtain the RRE, we set $g_i(n) = \Delta s_{n+i-1}$ and $y_j = \Delta^2 s_{j-1}$ for $i, j = 1, \ldots, k$. As for the MMPE, we set $n = 0$ and $S_k^{(0)} = x_k$. Then, from (3.9) and (3.10), we obtain

$$\alpha_k^{(0)} = b - C x_k = r_k\,. \tag{3.26}$$

Coming back to the GPA and using (3.26), we get the following new algorithm:

$$\begin{aligned} g_{0,i} &= \Delta s_i, \quad h_{0,i} = \Delta^2 s_i, \qquad i = 1, \ldots, k\,, \\ x_k &= x_{k-1} - \delta_k g_{k-1,k}\,, \\ r_k &= r_{k-1} - \delta_k h_{k-1,k}\,, \\ g_{i,k+1} &= g_{i-1,k+1} - \delta_k^{(i)} g_{i-1,i}\,, \\ h_{i,k+1} &= h_{i-1,k+1} - \delta_k^{(i)} h_{i-1,i}\,, \qquad i = 1, \ldots, k\,, \end{aligned}$$

where

$$\delta_k = \frac{\langle \Delta^2 s_{k-1}, r_{k-1} \rangle}{\langle \Delta^2 s_{k-1}, h_{k-1,k} \rangle}\,, \tag{3.27}$$

and for $i = 1, \ldots, k$

$$\delta_k^{(i)} = \frac{\langle \Delta^2 s_{i-1}, h_{i-1,k+1} \rangle}{\langle \Delta^2 s_{i-1}, h_{i-1,i} \rangle} . \tag{3.28}$$

This algorithm will be called the GAP-RRE. Let us give now some properties of this algorithm.

THEOREM 2

Let d be the degree of the minimal polynomial of B for the vector r_0. Then, $\forall k \leqslant d$, we have

(a) $\langle r_{k-1}, \Delta^2 s_{j-1} \rangle = 0, \quad j = 1, \ldots, k-1;$
(b) $\langle r_{k-1}, \Delta^2 s_{k-1} \rangle = -\langle r_{k-1}, h_{k-1,k} \rangle;$
(c) $\langle h_{i-1,i} \Delta^2 s_{i-1} \langle = -\langle h_{i-1,i}, h_{i-1,i} \rangle;$
(d) $\langle h_{i,k+1}, \Delta^2 s_{i-1} \rangle = -\langle h_{i,k+1}, h_{i-1,i} \rangle, \quad i = 1, \ldots, k;$
(e) $\langle h_{i-1,i} h_{j-1,j} \rangle = 0, \quad i \neq j.$

Proof

The proofs come directly from the determinantal expressions of r_k and $g_{k,i}$. We omit the details.

□

Invoking (b), (c) and (d) of the previous theorem, we obtain another formulation of the coefficients δ_k and $\delta_k^{(i)}$,

$$\delta_k = \frac{\langle r_{k-1}, h_{k-1,k} \rangle}{\langle h_{k-1,k} h_{k-1,k} \rangle} \tag{3.29}$$

and

$$\delta_k^{(i)} = \frac{\langle h_{i,k+1}, h_{i-1,i} \rangle}{\langle h_{i-1,i}, h_{i-1,i} \rangle} . \tag{3.30}$$

Let us mention here that since $h_{i,k} = -Cg_{i,k}$ for $i < k$, it is not necessary to compute the $h_{i,k}$'s. However, in this case, the algorithm requires more operations.

An important comparison criterion is the risk of breakdown presented by different methods. As it is known, the GCR can break down when C is not positive real, i.e. when its symmetric part is not positive definite. We will show that the GPA-RRE cannot break down even for problems with indefinite symmetric parts unless it has already converged.

In fact, since $\Delta s_0, \ldots, \Delta s_{i-1}$ are linearly independent for $i \leqslant d$, where d is the degree of the minimal polynomial of B for the vector $r_0, g_{i-1,i}$ do not vanish and then $h_{i-1,i} = -Cg_{i-1,i} \neq 0$. Thus, using the formulations (3.29) and (3.30) of δ_k and $\delta_k^{(i)}$, one can see that the GPA-RRE cannot break down.

The computation of x_k by the GPA-RRE requires $(3k+4)p+pZ$ multiplications per step. pZ denotes the number of nonzero elements in the matrix C.

Next, we shall briefly show how to derive the GCR from the GPA-RRE. Using some manipulations with determinants appearing in the expression (3.10), we can easily show that r_k can be expressed as

$$r_k = d_k \Delta s_k + \sum_{i=0}^{k-1} d_i^{(k)} \Delta s_i \,, \tag{3.31}$$

where

$$d_k = \frac{det(\Delta^2 S_k^{\mathrm{T}} \Delta S_k}{det(\Delta^2 S_k^{\mathrm{T}} \Delta^2 S_k)} \,.$$

On the other hand, $g_{k,k+1}$ can be developed as

$$g_{k,k+1} = \Delta s_k + \sum_{i=0}^{k-1} a_i^{(k)} \Delta s_i \,. \tag{3.32}$$

Now, assume that $d_k \neq 0$ and set $p_k = d_k g_{k,k+1}$. Then, from (3.31) and (3.32), we obtain

$$p_k - r_k \in span\{\Delta s_0, \ldots, \Delta s_{k-1}\} = span\{p_o, \ldots, p_{k-1}\} \,,$$

therefore we can write

$$p_k - r_k = \sum_{i=0}^{k-1} \eta_i^{(k)} p_i \,. \tag{3.33}$$

Using the result (e) of theorem 2 and the fact that $h_{i,k} = -Cg_{i,k}$ for $i<k$, it follows from (3.33) that

$$\eta_i^{(k)} = -\frac{\langle Cp_i, Cr_k \rangle}{\langle Cp_i, Cp_i \rangle} \,.$$

Replacing in the GPA-RRE, we obtain the following algorithm

$$\begin{cases} p_0 = r_0 = b - Cx_0 \quad \text{for } k = 0, 1, 2, \ldots \,; \\ x_{k+1} = x_k - \delta_k' p_k \,; \\ r_{k+1} = r_k + \delta_k' C p_k \,; \\ p_{k+1} = r_{k+1} + \sum_{i=0}^{k} \eta_i^{(k+1)} p_i \,, \end{cases}$$

where

$$\delta'_k = \frac{\delta_{k+1}}{d_k} = \frac{\langle r_k, Cp_k \rangle}{\langle Cp_k, Cp_k \rangle} .$$

Then we obtain exactly the GCR algorithm as described in [7]. In [21], Sidi showed that the RRE is equivalent to the GCR. We gave here another proof of this result and showed how the GCR could be obtained from the GPA-RRE.

Before closing this section, let us mention that we can define similarly the GPA-MPE which could be connected to the method of Arnoldi described by Saad in [19].

4. The Sβ algorithm

We noticed earlier that for the MMPE, the vector $S_k^{(n)}$ can be computed by the Sβ algorithm. In [15] we gave the algebraic properties of this algorithm and showed that it can be used for the implementation of the method of Henrici [13] for solving nonlinear systems. It can also be used for computing the eigenelements of a matrix [14].

In this section, we apply the Sβ for solving systems of linear equations and show that it is theoretically equivalent to the Orthodir and to the GPA-MMPE.

Let $(s_n)_{n\geqslant 0}$ be a sequence of vectors of $\mathbb{R}^p$, and $\{y_1, \ldots, y_k\}$ a set of linearly independent vectors of $\mathbb{R}^p$. We define the vector $S_k^{(n)}$ as a ratio of two determinants

$$S_k^{(n)} = \frac{\begin{vmatrix} s_n & s_{n+1} & \cdots & s_{n+k} \\ \langle y_1, \Delta s_n \rangle & \langle y_1, \Delta s_{n+1} \rangle & \cdots & \langle y_1, \Delta s_{n+k} \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle y_k, \Delta s_n \rangle & \langle y_k, \Delta s_{n+1} \rangle & \cdots & \langle y_k, \Delta s_{n+k} \rangle \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \langle y_1, \Delta s_n \rangle & \langle y_1, \Delta s_{n+1} \rangle & \cdots & \langle y_1, \Delta s_{n+k} \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle y_k, \Delta s_n \rangle & \langle y_k, \Delta s_{n+1} \rangle & \cdots & \langle y_k, \Delta s_{n+k} \rangle \end{vmatrix}} . \tag{4.1}$$

Let $\beta_k^{(n)}$ be the vector obtained from $S_k^{(n)}$ be replacing the first row in the numerator of (4.1) by $\Delta s_n, \Delta s_{n+1}, \ldots, \Delta s_{n+k}$.

For various values of n and k, $S_k^{(n)}$ can be recursively computed by the Sβ algorithm defined as follows:

$$\begin{cases} S_0^{(n)} = s_n, \quad \beta_0^{(n)} = \Delta s_n, \quad n \geqslant 0 , \\ S_k^{(n)} = S_{k-1}^{(n)} - \mu_k^{(n)} \Delta S_{k-1}^{(n)} , \\ \beta_k^{(n)} = \beta_{k-1}^{(n)} - \mu_k^{(n)} \Delta \beta_{k-1}^{(n)} , \end{cases} \tag{4.2}$$

where

$$\mu_k^{(n)} = \frac{\langle y_k, \beta_{k-1}^{(n)} \rangle}{\langle y_k, \Delta\beta_{k-1}^{(n)} \rangle}. \tag{4.3}$$

For solving the linear system (3.6), we shall consider now the case where the sequence (s_n) is generated linearly by (3.4). Using the fact that $\Delta s_n = b - Cs_n$, we obtain

$$\beta_k^{(n)} = b - CS_k^{(n)},$$

hence

$$\Delta\beta_k^{(n)} = \beta_k^{(n+1)} - \beta_k^{(n)} = -C\Delta S_k^{(n)}. \tag{4.4}$$

Taking $n = 0$ and setting $x_k = S_k^{(0)}$, we have

$$\beta_k^{(0)} = b - Cx_k = r_k.$$

In what follows, we shall show how to obtain the Orthodir method from the Sβ algorithm. Let us set

$$q_k = (-1)^k \Delta S_k^{(0)}, \quad k = 0, 1, \ldots. \tag{4.5}$$

Therefore, using (4.4) and (4.5) in (4.2), we obtain the relations

$$x_k = x_{k-1} + c_k q_{k-1}, \tag{4.6}$$

$$r_k = r_{k-1} - c_k C q_{k-1}, \tag{4.7}$$

where

$$c_k = (-1)^k \mu_k^{(0)} = \frac{\langle y_k, r_{k-1} \rangle}{\langle y_k, Cq_{k-1} \rangle}, \tag{4.8}$$

and where the direction q_k is computed by the Sβ algorithm. Now if we replace in the two determinants appearing in the determinantal expression of $S_k^{(n)}$, each column by its difference with the next one, we get

$$S_k^{(n)} = (-1)^k s_{k+n} + \sum_{i=0}^{k-1} a_{i,k}^{(n)} \Delta s_{n+i} \tag{4.9}$$

and similarly

$$\beta_k^{(n)} = (-1)^k \Delta s_{k+n} + \sum_{i=0}^{k-1} a_{i,k}^{(n)} \Delta^2 s_{n+i}. \tag{4.10}$$

Taking $n = 0$ and $n = 1$ in (4.9) and (4.10) and using (4.5), we can easily show that

$$q_k - Cq_{k-1} \in span\{\Delta s_0, \ldots, \Delta s_{k-1}\} = span\{q_0, \ldots, q_{k-1}\} \tag{4.11}$$

and consequently

$$q_k = Cq_{k-1} + \sum_{i=0}^{k-1} \gamma_k^{(k)} q_i \,. \tag{4.12}$$

We wish now to calculate the coefficients $\gamma_i^{(k)}$. Multiplying on the left by the matrix C and by the vector $y_j, j = 1, \ldots, k$, we get the following linear system

$$\langle y_j, Cq_k\rangle = \langle y_j, C^2 q_{k-1}\rangle + \sum_{i=0}^{k-1} \gamma_i^{(k)} \langle y_j, Cq_i\rangle, \quad j = 1, \ldots, k \,. \tag{4.13}$$

From the definition (4.5) of q_i we have

$$\langle y_j, Cq_i\rangle = (-1)^{i+1} \langle y_j, \Delta\beta_i^{(0)}\rangle \,.$$

But since $\langle y_j, \beta_i^{(1)}\rangle = \langle y_j, \beta_i^{(0)}\rangle = 0$ for $i = 1, \ldots, j$, it follows that

$$\langle y_j, Cq_i\rangle = 0, \quad i = 1, \ldots, j.$$

This last relation shows that (4.13) is a lower triangular system with the unknowns $\gamma_0^{(k)}, \ldots, \gamma_{k-1}^{(k)}$ given by

$$\gamma_o^{(k)} = -\frac{\langle y_1, C^2 q_{k-1}}{\langle y_1, Cq_0\rangle} \,, \tag{4.14}$$

and for $i = 1, \ldots, k-1$

$$\gamma_i^{(k)} = \frac{-\langle y_{i+1}, C^2 q_{k-1}\rangle - \sum_{j=0}^{i-1} \gamma_j^{(k)} \langle y_{i+1}, Cq_j\rangle}{\langle y_{i+1}, Cq_i\rangle} \,. \tag{4.15}$$

Thus the algorithm obtained here is exactly the algorithm (3.25) from which we derived the Orthodir and the orthogonal residual methods.

Acknowledgements

The author would like to thank the referee for his useful comments and valuable remarks.

References

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

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

[3] C. Brezinski, Other manifestations of the Schur complement, Lin. Alg. Appl. 24 (1988) 231–247.

[4] C. Brezinski and H. Sadok, Vector sequence transformation and fixed point methods, in: *Numerical Methods in Laminar and Turbulent Flows*, eds. C. Taylor et al. (Pineridge, Swansea, 1987) pp. 3–11.

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

[6] R.P. Eddy, Extrapolation to the limit of a vector sequence, in: *Information Linkage Between Applied Mathematics and Industry*, ed. P.C.C. Wang (Academic Press, New York, 1979) pp. 387–396.

[7] S.C. Eisenstat, H.C. Elman and M.H. Schultz, Variational iterative methods for nonsymmetric systems of linear equations, SIAM J. Numer. Anal. 20 (1983) 345–357.

[8] H.C. Elman, Iterative methods for large sparse nonsymmetric systems of linear equations, Ph.D. thesis, Computer Science Dept., Yale Univ., New Haven, CT (1982).

[9] V. Faber and T. Manteufel, Orthogonal Error Methods, SIAM J. Numer. Anal. 24 (1987) 170–187.

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

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

[12] B. Germain-Bonne, Estimation de la limite de suites et formalisation de procédés d'accélération de la convergence, Thèse d'Etat, Université de Lille 1 (1978).

[13] Henrici, *Elements of Numerical Analysis* (Wiley, New York, 1964).

[14] K. Jbilou, Méthodes d'extrapolation et de projection. Applications aux suites de vecteurs, Thèse de 3eme cycle, Université de Lille 1 (1988).

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

[16] K. Jbilou and H. Sadok, Analysis of the complete and incomplete vector extrapolation methods for solving systems of linear equations, submitted.

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

[18] 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.

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

[20] H. Sadok, Accélération de la convergence de suites vectorielles et méthodes de point fixe, Thèse, Univ. Lille (1988).

[21] A. Sidi, Extrapolation vs. projection methods for linear systems of equations, J. Comput. Appl. Math. 22 (1988) 71–88.

[22] A. Sidi, Convergence and stability of minimal polynomial and reduced rank extrapolation algorithms, SIAM J. Numer. Anal. 23 (1986) 197–209.

[23] A. Sidi, W.F. Ford and D.A. Smith, Acceleration of convergence of vector sequences, SIAM J. Numer. Anal. 23 (1986) 178–196.

[24] D.A. Smith, W.F. Ford and A. Sidi, Extrapolation methods for vector sequences, SIAM Rev. 29 (1987) 199-233; Correction, SIAM Rev. 30 (1988) 623–624.

[25] D.M. Young and K.C. Jea, Generalized conjugate-gradient acceleration of nonsymmetrisable iterative methods, Lin. Alg. Appl. 34 (1980) 159–194.