Numerical Algorithms 1 (1991) 261–284

AVOIDING BREAKDOWN AND NEAR-BREAKDOWN IN LANCZOS TYPE ALGORITHMS

C. BREZINSKI [1], M. REDIVO ZAGLIA [2] and H. SADOK [1]

[1] *Laboratoire d'Analyse Numérique et d'Optimisation, UFR IEEA - M3, Université des Sciences et Techniques de Lille Flandres-Artois, F-59655 Villeneuve d'Ascq Cedex, France*

[2] *Dipartimento di Elettronica e Informatica, Università degli Studi di Padova, via Gradenigo 6/a, I-35131 Padova, Italy*

Received 2 May 1991

Lanczos type algorithms form a wide and interesting class of iterative methods for solving systems of linear equations. One of their main interest is that they provide the exact answer in at most n steps where n is the dimension of the system. However a breakdown can occur in these algorithms due to a division by a zero scalar product. After recalling the so-called method of recursive zoom (MRZ) which allows to jump over such breakdown we propose two new variants. Then the method and its variants are extended to treat the case of a near-breakdown due to a division by a scalar product whose absolute value is small which is the reason for an important propagation of rounding errors in the method. Programming the various algorithms is then analyzed and explained. Numerical results illustrating the processes are discussed. The subroutines corresponding to the algorithms described can be obtained via *netlib*.

Subject classifications: AMS(MOS) 65F10, 65F25.

Keywords: Lanczos method, biconjugate gradient, projection, orthogonal polynomials, method of moments.

1. Lanczos type algorithms

Let us consider in C^n the system of linear equations

$$Ax = b.$$

Let $E_k = \text{span}(r_0, \ldots, A^{k-1}r_0)$ and $F_k = \text{span}(y, A^*y, \ldots, A^{*k-1}y)$ where r_0 and y are two non-zero arbitrary vectors and where A^* is the conjugate transpose of A. Let x_k be defined by

$$A_k(x_k - x_0) = -r_0 = -(Ax_0 - b)$$

where $A_k = H_k A H_k$, H_k being the oblique projection on E_k along $F_k^{\perp}$.

The residual vector $r_k = Ax_k - b$ satisfies

$$r_k = P_k(A) r_0$$

where P_k is a polynomial of degree at most k such that $P_k(0) = 1$ and where

$$\left(A^{*i}y, r_k\right) = 0 \quad \text{for } i = 0, \ldots, k-1.$$

If we set $c_i = (y, A^i r_0)$ and if we define the linear functional c on the space of polynomials by $c(\xi^i) = c_i$ for $i = 0, 1, \ldots$ then the preceding relations can be written as

$$c(\xi^i P_k(\xi)) = 0 \quad \text{for } i = 0, \ldots, k-1$$

which shows that P_k is the polynomial of degree at most k belonging to the family of formal orthogonal polynomials with respect to c.

A Lanczos type method (and among them the biconjugate gradient method of Fletcher [7]) consists in computing P_k recursively, then r_k and finally x_k such that $r_k = Ax_k - b$ without, of course, inverting A. Such a method gives the exact solution of the system in at most n steps, see [4] for a review of such methods.

The recursive computation of P_k involves the computation of some scalar products which appear as the denominators of the coefficients of the recurrence relations. Thus, if one of these scalar products is zero, then a breakdown occurs in the algorithm which has to be stopped. This is due to the non-existence of some of the polynomials P_k and the breakdown can be avoided by jumping over these polynomials and computing only the existing ones. The corresponding method was presented in [3] with its derivation and theoretical background. It was called the *Method of Recursive Zoom*, in short the MRZ. This method will be recalled in the next section and some new variants will be presented in section 3. Let us mention that breakdown can be similarly avoided in the conjugate gradient squared (CGS) method [9], see [5].

If the scalar products in the denominators are different from zero but small in absolute value, an important propagation of rounding errors could occur in the algorithm, a situation known as a near-breakdown. In section 4 we shall present some generalizations of the MRZ for treating this problem. Programming the various algorithms will be analyzed in section 5. Numerical results will be discussed in the last section. The subroutines (in FORTRAN 77) corresponding to the algorithms described in this paper can be obtained via *netlib*.

2. The MRZ

As explained in the preceding section we shall consider only the existing orthogonal polynomials (called regular). Thus let P_k be the polynomial of degree at most n_k satisfying the orthogonality relations

$$c(\xi^i P_k(\xi)) = 0 \quad \text{for } i = 0, \ldots, n_k - 1$$

and normalized by the condition $P_k(0) = 1$.

Let $P_k^{(1)}$ be the monic polynomial of degree n_k belonging to the family of formal orthogonal polynomials with respect to the functional $c^{(1)}$ defined by $c^{(1)}(\xi^i) = c(\xi^{i+1})$.

If $P_k^{(1)}$ satisfies

$$c^{(1)}\left(\xi^i P_k^{(1)}(\xi)\right)=0 \quad \text{for } i=0,\ldots,n_k+m_k-2$$

and

$$c^{(1)}\left(\xi^{n_k+m_k-1} P_k^{(1)}(\xi)\right)\neq 0$$

then, it was proved in [3] that

$$P_{k+1}(\xi)=P_k(\xi)-\xi w_k(\xi)P_k^{(1)}(\xi)$$
$$P_{k+1}^{(1)}(\xi)=q_k(\xi)P_k^{(1)}(\xi)-C_{k+1}P_{k-1}^{(1)}(\xi)$$

with $P_{-1}^{(1)}(\xi)=0$, $P_0^{(1)}(\xi)=1$ and $C_1=0$. w_k is a polynomial of degree m_k-1 at most and q_k is a monic polynomial of degree m_k. Their coefficients and C_{k+1} are determined by imposing the orthogonality relations of P_{k+1} and $P_{k+1}^{(1)}$.

If we set

$$r_k=P_k(A)r_0$$
$$z_k=P_k^{(1)}(A)r_0$$

then the preceding recurrence relations give

$$r_{k+1}=r_k-Aw_k(A)z_k$$
$$x_{k+1}=x_k-w_k(A)z_k$$
$$z_{k+1}=q_k(A)z_k-C_{k+1}z_{k-1}$$

with $z_0=r_0$, $z_{-1}=0$ and $C_1=0$.

m_k is determined by the conditions

$$\left(y,\ A^{i+1}z_k\right)=0, \quad \text{for } i=0,\ldots,n_k+m_k-2$$

and

$$\left(y,\ A^{n_k+m_k}z_k\right)\neq 0.$$

If we set $w_k(\xi)=\beta_0+\cdots+\beta_{m_k-1}\xi^{m_k-1}$ then the β's are given by

$$\begin{cases}\beta_{m_k-1}\left(y,\ A^{n_k+m_k}z_k\right)=\left(y,\ A^{n_k}r_k\right)\\ \beta_{m_k-2}\left(y,\ A^{n_k+m_k}z_k\right)+\beta_{m_k-1}\left(y,\ A^{n_k+m_k+1}z_k\right)=\left(y,\ A^{n_k+1}r_k\right)\\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\ \beta_0\left(y,\ A^{n_k+m_k}z_k\right)+\beta_1\left(y,\ A^{n_k+m_k+1}z_k\right)+\cdots+\beta_{m_k-1}\left(y,\ A^{n_k+2m_k-1}z_k\right)\\ \quad=\left(y,\ A^{n_k+m_k-1}r_k\right).\end{cases}$$

If we set $q_k(\xi)=\alpha_0+\cdots+\alpha_{m_k-1}\xi^{m_k-1}+\xi^{m_k}$ then we have

$$\begin{cases}C_{k+1}\left(y,\ A^{n_k}z_{k-1}\right)=\left(y,\ A^{n_k+m_k}z_k\right)\\ \alpha_{m_k-1}\left(y,\ A^{n_k+m_k}z_k\right)=C_{k+1}\left(y,\ A^{n_k+1}z_{k-1}\right)-\left(y,\ A^{n_k+m_k+1}z_k\right)\\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\ \alpha_0\left(y,\ A^{n_k+m_k}z_k\right)+\alpha_1\left(y,\ A^{n_k+m_k+1}z_k\right)+\cdots+\alpha_{m_k-1}\left(y,\ A^{n_k+2m_k-1}z_k\right)\\ \quad=C_{k+1}\left(y,\ A^{n_k+m_k}z_{k-1}\right)-\left(y,\ A^{n_k+2m_k}z_k\right).\end{cases}$$

This method has been called the *Method of Recursive Zoom*, in short the MRZ. It can only present an incurable hard breakdown if $(y, A^n z_k) = 0$ where n is the dimension of the system to be solved.

3. Variants of the MRZ

In the MRZ the polynomial $P_{k+1}^{(1)}$ is computed from $P_k^{(1)}$ and $P_{k-1}^{(1)}$ by the three-term recurrence relationship.

We shall now see how to compute $P_{k+1}^{(1)}$ from $P_k^{(1)}$ and from either P_k or P_{k+1}.

In the first case the variant has been called the SMRZ, where S stands for *symmetric*, since P_{k+1} is also computes from the same polynomials thus leading to a symmetry between both relations.

In the second case the variant has been called the BMRZ, where B stands for *balancing*, since the method computes P_{k+1} from $P_k^{(1)}$ and P_k by its first relation, then $P_{k+1}^{(1)}$ from P_{k+1} and $P_k^{(1)}$ by its second relation, then goes back to the first relation and so on, thus balancing between both relations.

In these variants the same letter has been often used to designate different objects but no confusion is possible.

3.1. THE SMRZ

In this variant, $P_{k+1}^{(1)}$ shall be expressed as

$$P_{k+1}^{(1)}(\xi) = t_k(\xi) P_k^{(1)}(\xi) - D_{k+1} P_k(\xi)$$

where t_k is a monic polynomial of degree m_k and D_{k+1} is a constant.

We set

$$t_k(\xi) = \delta_0 + \cdots + \delta_{m_k-1}\xi^{m_k-1} + \xi^{m_k}.$$

We have

$$c^{(1)}(\xi^i P_{k+1}^{(1)}) = \delta_0 c^{(1)}(\xi^i P_k^{(1)}) + \cdots + \delta_{m_k-1} c^{(1)}(\xi^{i+m_k-1} P_k^{(1)}) + c^{(1)}(\xi^{i+m_k} P_k^{(1)})$$
$$- D_{k+1} c(\xi^{i+1} P_k).$$

P_{k+1} must satisfy the orthogonality conditions

$$c^{(1)}(\xi^i P_{k+1}^{(1)}) = 0 \quad \text{for } i = 0, \ldots, n_k + m_k - 1.$$

But $c^{(1)}(\xi^{i+m_k} P_k^{(1)}) = 0$ for $i = 0, \ldots, n_k - 2$ and $c(\xi^{i+1} P_k) = 0$ for $i = 0, \ldots, n_k - 2$. Thus the orthogonality conditions of $P_{k+1}^{(1)}$ are satisfied for $i = 0, \ldots, n_k - 2$. Imposing these conditions for $i = n_k - 1, \ldots, n_k + m_k - 1$ leads to $m_k + 1$ relations for determining the $m_k + 1$ coefficients $\delta_0, \ldots, \delta_{m_k-1}$ and D_{k+1}.

These conditions give

for $i = n_k - 1$

$$D_{k+1} c(\xi^{n_k} P_k) = c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)})$$

for $i = n_k$

$$D_{k+1} c(\xi^{n_k+1} P_k) - \delta_{m_k-1} c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) = c^{(1)}(\xi^{n_k+m_k} P_k^{(1)})$$

. . . .

for $i = n_k + m_k - 1$

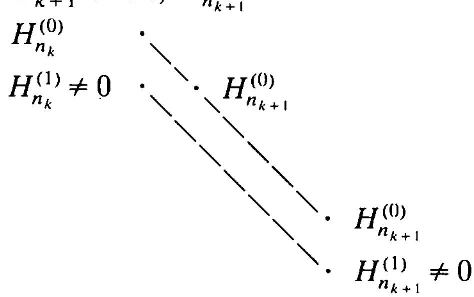
$$D_{k+1} c(\xi^{n_k+m_k} P_k) - \delta_0 c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) - \cdots - \delta_{m_k-1} c^{(1)}(\xi^{n_k+2m_k-2} P_k^{(1)}) = c^{(1)}(\xi^{n_k+2m_k-1} P_k^{(1)}).$$

Thus the first equation gives D_{k+1}, the second one $\delta_{m_k-1}, \ldots$, and the last one δ_0.

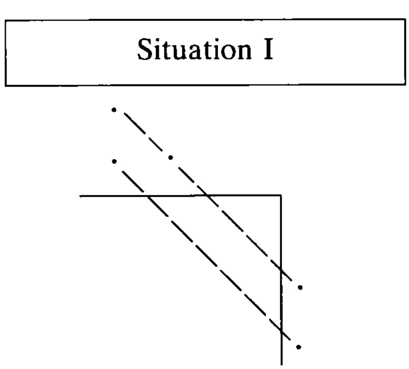
Since $c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) \neq 0$ this system is regular if and only if $c(\xi^{n_k} P_k) \neq 0$. Let us now look at this condition. We have

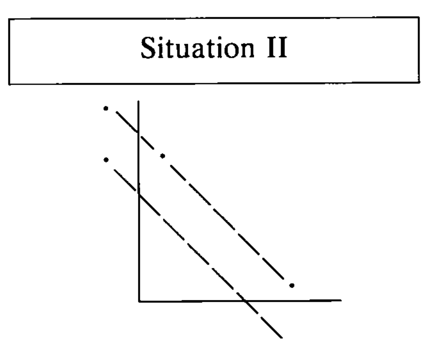
$$c(\xi^{n_k} P_k) = (-1)^{n_k} \frac{H_{n_k+1}^{(0)}}{H_{n_k}^{(1)}}.$$

Thus $c(\xi^{n_k} P_k) = 0$ if and only if $H_{n_k+1}^{(0)} = 0$. Since $P_k^{(1)}$ exists, $H_{n_k}^{(1)} \neq 0$ and since $P_{k+1}^{(1)}$ exists, $H_{n_{k+1}}^{(1)} \neq 0$. Thus we have



and we are in one of the situations (see Draux [6], property 1.9, p. 25)





In I, $H_{n_k+1}^{(0)} \neq 0$ while, in II, $H_{n_k+1}^{(0)} = 0$ and $H_{n_k}^{(0)} \neq 0$.

Thus, if $c(\xi^{n_k} P_k) = 0$ we are in the situation II. It follows that the monic orthogonal polynomial of degree n_k with respect to c exists. Let us call it $P_k^{(0)}$. We have

$$c\left(\xi^{n_k} P_k^{(0)}\right) = 0.$$

We also have

$$c\left(P_k^{(1)}\right) = (-1)^{n_k} \frac{H_{n_k+1}^{(0)}}{H_{n_k}^{(1)}} = 0.$$

But $c(P_k^{(1)}) = c^{(1)}(\xi^{-1} P_k^{(1)}) = 0$ which shows (Draux [6], corollary 2.2, p. 137) that the polynomial $P_k^{(1)}$ is on the west side of a square block but not at its north-west corner. Thus $P_k^{(0)}$ is on the west side or at the north-west corner of a block and it is identical to $P_k^{(1)}$ (Draux [6], property 2.1, p. 100) which means that the orthogonal polynomials are in the situation II as the corresponding Hankel determinants. Moreover

$$c^{(1)}\left(\xi^i P_k^{(1)}\right) = c^{(1)}\left(\xi^i P_k^{(0)}\right) = c\left(\xi^{i+1} P_k^{(0)}\right) \begin{cases} = 0 & \text{for } i = 0, \ldots, n_k + m_k - 2 \\ \neq 0 & \text{for } i = n_k + m_k - 1. \end{cases}$$

Since $P_k^{(0)}$ is orthogonal regular $c(P_k^{(0)}) = 0$ and we finally have

$$c\left(\xi^i P_k^{(0)}\right) \begin{cases} = 0 & \text{for } i = 0, \ldots, n_k + m_k - 1 \\ \neq 0 & \text{for } i = n_k + m_k. \end{cases}$$

But

$$P_k(\xi) = (-1)^{n_k} \frac{H_{n_k}^{(0)}}{H_{n_k}^{(1)}} P_k^{(0)}(\xi)$$

and thus

$$c\left(\xi^i P_k\right) \begin{cases} = 0 & \text{for } i = 0, \ldots, n_k + m_k - 1 \\ \neq 0 & \text{for } i = n_k + m_k. \end{cases}$$

But, by the first relation of the MRZ

$$P_{k+1}(\xi) = P_k(\xi) - \xi w_k(\xi) P_k^{(1)}(\xi)$$

where w_k is a polynomial of degree $m_k - 1$. Setting

$$w_k(\xi) = \beta_0 + \cdots + \beta_{m_k-1} \xi^{m_k-1}$$

we have

$$c\left(\xi^i P_{k+1}\right) = c\left(\xi^i P_k\right) - \beta_0 c^{(1)}\left(\xi^i P_k^{(1)}\right) - \cdots - \beta_{m_k-1} c^{(1)}\left(\xi^{i+m_k-1} P_k^{(1)}\right).$$

But $c^{(1)}(\xi^{i+m_k-1} P_k^{(1)}) = 0$ for $i = 0, \ldots, n_k - 1$ and $c(\xi^i P_k) = 0$ for $i = 0, \ldots, n_k + m_k - 1$ and it follows that $\beta_0 = \cdots = \beta_{m_k-1} = 0$. Thus we have proved the

THEOREM

$c(\xi^{n_k}P_k)=0$ if and only if P_{k+1} is identical to P_k.

If $c(\xi^{n_k}P_k)=0$ the relation of the SMRZ cannot be used and we shall look for a relation of the form

$$P_{k+1}^{(1)}(\xi)=t_k(\xi)P_k^{(1)}(\xi)-u_k(\xi)P_k(\xi)$$

where t_k and u_k are polynomials of degree m_k at most. Thus we have $2m_k+2$ unknown coefficients to determine.

Moreover

$$c^{(1)}\left(\xi^i t_k P_k^{(1)}\right)=0 \quad \text{for } i=0,\ldots,n_k-2$$

and

$$c\left(\xi^{i+1}u_k P_k\right)=0 \quad \text{for } i=0,\ldots,n_k-2.$$

Thus $c^{(1)}(\xi^i P_{k+1}^{(1)})=0$ for $i=0,\ldots,n_k-2$. Imposing this condition for $i=n_k-1,\ldots,n_k+m_k-1$ leads to m_k+1 relations. Imposing that $P_{k+1}^{(1)}$ be monic gives one more relation and thus we have m_k+2 relations for $2m_k+2$ unknowns. Let us give a numerical example showing that this system is inconsistent

We take $c_0=1$, $c_1=2$, $c_2=4$, $c_3=c_4=8$, $c_5=c_6=0$.

We have $H_2^{(0)}=H_2^{(1)}=H_3^{(0)}=0$ and $H_2^{(2)}=-32$.

Thus $n_1=1$, $m_1=2$ and $n_2=3$.

We find that

$$P_1^{(0)}(\xi)=P_1^{(1)}(\xi)=P_1^{(2)}(\xi)=\xi-2$$

$$P_1(\xi)=1-\xi/2$$

$$P_1^{(3)}(\xi)=\xi-1$$

$$P_2^{(1)}(\xi)=\xi^3-2\xi^2+4\xi-4$$

$$P_2(\xi) \quad \text{is identical to } P_1(\xi)$$

We shall try to express $P_2^{(1)}$ as a combination of $P_1^{(1)}$ and P_1 that is

$$P_2^{(1)}(\xi)=(\xi-2)(a\xi^2+b\xi+c)+(1-\xi/2)(a'\xi^2+b'\xi+c')$$

which leads to

$$2a-a'=2$$

$$2b-2(2a-a')-b'=-4$$

$$2c-2(2b-b')-c'=8$$

$$2c-c'=4.$$

From the first two equations we obtain $2b-b'=0$. Replacing in the third one leads to $2c-c'=8$ which is inconsistent with the last equation.

Thus the SMRZ can only be used if $c(\xi^{n_k}P_k)\neq 0$. If this is not the case $P_{k+1}^{(1)}$ cannot be expressed, in general, as a combination (with polynomial coefficients) of $P_k^{(1)}$ and P_k.

This is the reason why we shall now try to express $P_{k+1}^{(1)}$ as a combination of $P_k^{(1)}$ and P_{k+1}.

3.2. THE BMRZ

In this variant, $P_{k+1}^{(1)}$ shall be expressed as

$$P_{k+1}^{(1)}(\xi) = A_{k+1} P_{k+1}(\xi) + B_{k+1} P_k^{(1)}(\xi)$$

where A_{k+1} and B_{k+1} are constants.

We have

$$c^{(1)}(\xi^i P_{k+1}^{(1)}) = A_{k+1} c(\xi^{i+1} P_{k+1}) + B_{k+1} c^{(1)}(\xi^i P_k^{(1)}).$$

Since $c^{(1)}(\xi^i P_k^{(1)}) = 0$ for $i = 0, \ldots, n_k + m_k - 2$ and $c(\xi^{i+1} P_{k+1}) = 0$ for $i = 0, \ldots, n_k + m_k - 2$, the orthogonality relations of $P_{k+1}^{(1)}$ are also satisfied for these indexes. Writing this condition for $i = n_k + m_k - 1$ gives

$$A_{k+1} c(\xi^{n_k+m_k} P_{k+1}) + B_{k+1} c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) = 0.$$

Let us now impose that $P_{k+1}^{(1)}$ is monic. Since $P_{k+1}^{(1)}$ has the exact degree $n_k + m_k$ the relation of the BMRZ can only hold if P_{k+1} has the exact degree $n_k + m_k$, that is if $H_{n_{k+1}}^{(0)} \neq 0$ or, in other terms, if we are in the situation I described in the SMRZ. From the first relation of the MRZ the coefficient of $\xi^{n_k+m_k}$ in P_{k+1} is equal to $-\beta_{m_k-1}$. Thus we must take

$$A_{k+1} = -\frac{1}{\beta_{m_k-1}}$$

and it follows that

$$B_{k+1} = \frac{1}{\beta_{m_k-1}} \cdot \frac{c(\xi^{n_k+m_k} P_{k+1})}{c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)})} = \frac{c(\xi^{n_k+m_k} P_{k+1})}{c(\xi^{n_k} P_k)}.$$

This relation is, in fact, the same as that given by Draux [6] (pp. 397–398) since $P_{k+1}^{(0)}$ exists because $H_{n_{k+1}}^{(0)} \neq 0$. It is a generalization of the second relation between adjacent families of orthogonal polynomials (that is the relation involving $e_k^{(0)}$, see Brezinski [2] (relation 2.9, p. 92 or p. 95)).

Setting, as in the MRZ, $r_k = P_k(A) r_0$ and $z_k = P_k^{(1)}(A) r_0$ we recover the usual formulation of the biconjugate gradient method of Fletcher [7], see also Brezinski [2] p. 91, in which z_{k+1} is computed from r_{k+1} and z_k.

If we are in the situation II described in the SMRZ then P_{k+1} is identical to P_k and $P_{k+1}^{(1)}$ cannot be computed from $P_k^{(1)}$ and P_k even if the constants A_{k+1} and B_{k+1} are replaced by polynomials.

The BMRZ is, in fact, a simplified (and simplest) version of the SMRZ since if P_{k+1} is replaced by

$$P_{k+1}(\xi) = P_k(\xi) - \xi w_k(\xi) P_k^{(1)}(\xi)$$

in the BMRZ we obtain

$$P_{k+1}^{(1)}(\xi) = A_{k+1} P_k(\xi) + \left(B_{k+1} - A_{k+1}\xi w_k(\xi)\right) P_k^{(1)}(\xi)$$

which is the SMRZ since, in this method, t_k and D_{k+1} are uniquely determined. However since, in both methods, the computations are not conducted in the same way, their numerical stability has to be compared.

4. Near-breakdown

As explained in [3], a breakdown occurs in a Lanczos type method when

$$c^{(1)}\left(\xi^{n_k} P_k^{(1)}\right) = 0.$$

In that case the monic orthogonal polynomial of degree $n_k + 1$ with respect to $c^{(1)}$ does not exist.

It is possible to avoid such a breakdown by using the first regular orthogonal polynomial with respect to $c^{(1)}$ following $P_k^{(1)}$. This polynomial was denoted by $P_{k+1}^{(1)}$, its degree was $n_{k+1} = n_k + m_k$ where m_k is determined such that

$$\begin{aligned} c^{(1)}\left(\xi^i P_k^{(1)}\right) &= 0 \quad \text{for } i = 0, \ldots, n_k + m_k - 2 \\ &\neq 0 \quad \text{for } i = n_k + m_k - 1. \end{aligned}$$

$P_{k+1}^{(1)}$ and P_{k+1} are then computed recursively by two relations whose coefficients are given as solutions of a triangular system of linear equations with $c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)})$ on its diagonal.

Of course if $|c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)})|$ is different from zero but small (and possibly badly computed) the coefficients of the two recurrence relations of the MRZ(or its variants) could be large and badly computed and thus rounding errors could affect the algorithm. The same is true if the quantities $|c^{(1)}(\xi^i P_k^{(1)})|$ are not zero for $i = n_k, \ldots, n_k + m_k - 2$ but small; in that case no breakdown occurs in the method but numerical instability could be present, a situation called near-breakdown.

It is possible to avoid such a near-breakdown by jumping over those polynomials which could be badly computed and to compute directly the first regular polynomial following them. Thus, let $\varepsilon \geqslant 0$ be given. We define $m_k \geqslant 1$ such that

$$\left|c^{(1)}\left(\xi^i P_k^{(1)}\right)\right| \leqslant \varepsilon \quad \text{for } i = n_k, \ldots, n_k + m_k - 2$$

and

$$\left|c^{(1)}\left(\xi^i P_k^{(1)}\right)\right| > \varepsilon \quad \text{for } i = n_k + m_k - 1.$$

Let $n_{k+1} = n_k + m_k$. We shall denote by $P_{k+1}^{(1)}$ the regular orthogonal polynomial of degree n_{k+1} with respect to $c^{(1)}$. As explained in the sequel if such a polynomial does not exist (and we shall be able to detect such a case) the value of m_k has to be increased until a regular polynomial has been obtained. We shall denote by P_{k+1} the corresponding orthogonal polynomial of degree n_{k+1} at most with respect to c normalized by the condition $P_{k+1}(0)=1$.

Let us first compute P_{k+1}. We shall write it under the form

$$P_{k+1}(\xi)=P_k(\xi)-\xi w_k(\xi)P_k^{(1)}(\xi)-\xi v_k(\xi)P_k(\xi)$$

where w_k is a polynomial of degree m_k-1 at most and v_k a polynomial of degree m_k-2 at most. Let us recall that, in the case of a breakdown, v_k is identically zero.

We have

$$c\left(\xi^i P_{k+1}\right)=c\left(\xi^i P_k\right)-c^{(1)}\left(\xi^i w_k P_k^{(1)}\right)-c\left(\xi^{i+1} v_k P_k\right).$$

But we have

$$\begin{aligned} c\left(\xi^i P_k\right)&=0 \qquad i=0,\ldots,n_k-1\\ c^{(1)}\left(\xi^i w_k P_k^{(1)}\right)&=0 \qquad i=0,\ldots,n_k-m_k\\ c\left(\xi^{i+1} v_k P_k\right)&=0 \qquad i=0,\ldots,n_k-m_k. \end{aligned}$$

Thus $c(\xi^i P_{k+1})=0$ for $i=0,\ldots,n_k-m_k$. Writing that this condition is satisfied for $i=n_k-m_k+1,\ldots,n_k+m_k-1$ yields $2m_k-1$ relations for determining the $2m_k-1$ unknowns which are the m_k coefficients of w_k and the m_k-1 coefficients of v_k. Of course we must assume that $n_k-m_k+1\geqslant 0$. The case $n_k-m_k+1<0$ will be treated below.

We set

$$\begin{aligned} w_k(\xi)&=\beta_0+\cdots+\beta_{m_k-1}\xi^{m_k-1}\\ v_k(\xi)&=\beta_0'+\cdots+\beta_{m_k-2}'\xi^{m_k-2} \end{aligned}$$

Writing the orthogonality conditions of P_{k+1} gives

for $i=n_k-m_k+1$

$$\beta_{m_k-1}c^{(1)}\left(\xi^{n_k}P_k^{(1)}\right)+\beta_{m_k-2}'c\left(\xi^{n_k}P_k\right)=0$$

. . . .

for $i=n_k-1$

$$\begin{aligned} &\beta_1 c^{(1)}\left(\xi^{n_k}P_k^{(1)}\right)+\cdots+\beta_{m_k-1}c^{(1)}\left(\xi^{n_k+m_k-2}P_k^{(1)}\right)\\ &\quad+\beta_0' c\left(\xi^{n_k}P_k\right)+\cdots+\beta_{m_k-2}'c\left(\xi^{n_k+m_k-2}P_k\right)=0 \end{aligned}$$

for $i=n_k$

$$\begin{aligned} &\beta_0 c^{(1)}\left(\xi^{n_k}P_k^{(1)}\right)+\cdots+\beta_{m_k-1}c^{(1)}\left(\xi^{n_k+m_k-1}P_k^{(1)}\right)\\ &\quad+\beta_0' c\left(\xi^{n_k+1}P_k\right)+\cdots+\beta_{m_k-2}'c\left(\xi^{n_k+m_k-1}P_k\right)=c\left(\xi^{n_k}P_k\right) \end{aligned}$$

. . . .

for $i=n_k+m_k-1$

$$\begin{aligned} &\beta_0 c^{(1)}\left(\xi^{n_k+m_k-1}P_k^{(1)}\right)+\cdots+\beta_{m_k-1}c^{(1)}\left(\xi^{n_k+2m_k-2}P_k^{(1)}\right)\\ &\quad+\beta_0' c\left(\xi^{n_k+m_k}P_k\right)+\cdots+\beta_{m_k-2}'c\left(\xi^{n_k+2m_k-2}P_k\right)=c\left(\xi^{n_k+m_k-1}P_k\right). \end{aligned}$$

It must be noticed that, in the case of breakdown, that is when $c^{(1)}(\xi^i P_k^{(1)}) = 0$ for $i = n_k, \ldots, n_k + m_k - 2$, then the conditions for $i = n_k - m_k + 1, \ldots, n_k - 1$ give $\beta'_0 = \cdots = \beta'_{m_k - 2} = 0$ (even if $c(\xi^i P_k) = 0$ for $i = n_k, \ldots, n_k + m_k - 2$ since in this case the β'_is are arbitrary) and the conditions for $i = n_k, \ldots, n_k + m_k - 1$ reduce to those of the MRZ which is exactly recovered.

Let us now examine the case $n_k - m_k + 1 < 0$. We now have $n_k + m_k$ equations, with $n_k + m_k < 2m_k - 1$, for determining the $2m_k - 1$ unknown coefficients. This is the reason why we shall now take for v_k a polynomial of degree at most $n_k - 1$

$$v_k = \beta'_0 + \cdots + \beta'_{n_k - 1}\xi^{n_k - 1}.$$

The orthogonality conditions of P_{k+1} give

for $i = 0$

$$\beta_{n_k} c^{(1)}\left(\xi^{n_k} P_k^{(1)}\right) + \cdots + \beta_{m_k - 1} c^{(1)}\left(\xi^{m_k - 1} P_k^{(1)}\right) + \beta'_{n_k - 1} c\left(\xi^{n_k} P_k\right) = 0$$

. . . .

for $i = n_k - 1$

$$\beta_1 c^{(1)}\left(\xi^{n_k} P_k^{(1)}\right) + \cdots + \beta_{m_k - 1} c^{(1)}\left(\xi^{n_k + m_k - 2} P_k^{(1)}\right)$$
$$+ \beta'_0 c\left(\xi^{n_k} P_k\right) + \cdots + \beta'_{n_k - 1} c\left(\xi^{2n_k - 1} P_k\right) = 0$$

for $i = n_k$

$$\beta_0 c^{(1)}\left(\xi^{n_k} P_k^{(1)}\right) + \cdots + \beta_{m_k - 1} c^{(1)}\left(\xi^{n_k + m_k - 1} P_k^{(1)}\right)$$
$$+ \beta'_0 c\left(\xi^{n_k + 1} P_k\right) + \cdots + \beta'_{n_k - 1} c\left(\xi^{2n_k} P_k\right) = c\left(\xi^{n_k} P_k\right)$$

. . . .

for $i = n_k + m_k - 1$

$$\beta_0 c^{(1)}\left(\xi^{n_k + m_k - 1} P_k^{(1)}\right) + \cdots + \beta_{m_k - 1} c^{(1)}\left(\xi^{n_k + 2m_k - 2} P_k^{(1)}\right)$$
$$+ \beta'_0 c\left(\xi^{n_k + m_k} P_k\right) + \cdots + \beta'_{n_k - 1} c\left(\xi^{2n_k + m_k - 1} P_k\right) = c\left(\xi^{n_k + m_k - 1} P_k\right).$$

In the case of breakdown the conditions for $i = 0, \ldots, n_k - 1$ give $\beta'_0 = \cdots = \beta'_{n_k - 1} = 0$ and we recover exactly the MRZ again.

Let us now try to compute recursively $P_{k+1}^{(1)}$. They are several possibilities corresponding to the MRZ or its two variants, the SMRZ and the BMRZ.

4.1. THE GMRZ

We shall try to generalize (G stands for *general* in the name of the method) the three-term recurrence relationship between the polynomials $P_k^{(1)}$.

Let $P_{k-1}^{(1)}$ and $P_k^{(1)}$ be two successive regular orthogonal polynomials with respect to $c^{(1)}$. By successive we mean that all the polynomials of degree $n_{k-1} + 1, \ldots, n_k - 1$ do not exist. We shall try to determine $P_{k+1}^{(1)}$, which is the regular orthogonal polynomial of degree $n_{k+1} = n_k + m_k$ with respect to $c^{(1)}$ but not necessarily the successor of $P_k^{(1)}$ (which means that some regular polynomials of degree between n_k+1 and $n_{k+1}-1$ can exist) under the form

$$P_{k+1}^{(1)}(\xi)=q_k(\xi)P_k^{(1)}(\xi)+r_k(\xi)P_{k-1}^{(1)}(\xi)$$

where q_k is a monic polynomial of degree m_k and r_k a polynomial of degree at most m_k-1.

We must have

$$c^{(1)}\big(\xi^i P_{k+1}^{(1)}\big)=0 \quad \text{for } i=0,\ldots,n_k+m_k-1.$$

But

$$c^{(1)}\big(\xi^i q_k P_k^{(1)}\big)=0 \quad \text{for } i=0,\ldots,n_k-m_k-1$$

and

$$c^{(1)}\big(\xi^i r_k P_{k-1}^{(1)}\big)=0 \quad \text{for } i=0,\ldots,n_k-m_k-1.$$

Thus the orthogonality conditions of $P_{k+1}^{(1)}$ are satisfied for the same indexes. Imposing these conditions for $i=n_k-m_k,\ldots,n_k+m_k-1$ will lead to $2m_k$ relations for determining the $2m_k$ unknown coefficients of q_k and r_k. Of course we must assume that $n_k-m_k\geqslant 0$. The case $n_k-m_k<0$ will be treated below.

We set

$$q_k(\xi)=\alpha_0+\cdots+\alpha_{m_k-1}\xi^{m_k-1}+\xi^{m_k}$$
$$r_k(\xi)=\alpha_0'+\cdots+\alpha_{m_k-1}'\xi^{m_k-1}.$$

For $i=n_k-m_k,\ldots,n_k+m_k-1$ we must have

$$c^{(1)}\big(\xi^i P_{k+1}^{(1)}\big)=\alpha_0 c^{(1)}\big(\xi^i P_k^{(1)}\big)+\cdots+\alpha_{m_k-1}c^{(1)}\big(\xi^{i+m_k-1}P_k^{(1)}\big)+c^{(1)}\big(\xi^{i+m_k}P_k^{(1)}\big)$$
$$+\alpha_0' c^{(1)}\big(\xi^i P_{k-1}^{(1)}\big)+\cdots+\alpha_{m_k-1}' c^{(1)}\big(\xi^{i+m_k-1}P_{k-1}^{(1)}\big)=0.$$

We obtain

for $i=n_k-m_k$

$$c^{(1)}\big(\xi^{n_k}P_k^{(1)}\big)+\alpha_{m_k-1}' c^{(1)}\big(\xi^{n_k-1}P_{k-1}^{(1)}\big)=0$$

for $i=n_k-m_k+1$

$$\alpha_{m_k-1}c^{(1)}\big(\xi^{n_k}P_k^{(1)}\big)+c^{(1)}\big(\xi^{n_k+1}P_k^{(1)}\big)+\alpha_{m_k-2}' c^{(1)}\big(\xi^{n_k-1}P_{k-1}^{(1)}\big)$$
$$+\alpha_{m_k-1}' c^{(1)}\big(\xi^{n_k}P_{k-1}^{(1)}\big)=0$$

. . . .

for $i=n_k-1$

$$\alpha_1 c^{(1)}\big(\xi^{n_k}P_k^{(1)}\big)+\cdots+\alpha_{m_k-1}c^{(1)}\big(\xi^{n_k+m_k-2}P_k^{(1)}\big)+c^{(1)}\big(\xi^{n_k+m_k-1}P_k^{(1)}\big)$$
$$+\alpha_0' c^{(1)}\big(\xi^{n_k-1}P_{k-1}^{(1)}\big)+\cdots+\alpha_{m_k-1}' c^{(1)}\big(\xi^{n_k+m_k-2}P_{k-1}^{(1)}\big)=0$$

for $i=n_k$

$$\alpha_0 c^{(1)}\big(\xi^{n_k}P_k^{(1)}\big)+\cdots+\alpha_{m_k-1}c^{(1)}\big(\xi^{n_k+m_k-1}P_k^{(1)}\big)+c^{(1)}\big(\xi^{n_k+m_k}P_k^{(1)}\big)$$
$$+\alpha_0' c^{(1)}\big(\xi^{n_k}P_{k-1}^{(1)}\big)+\cdots+\alpha_{m_k-1}' c^{(1)}\big(\xi^{n_k+m_k-1}P_{k-1}^{(1)}\big)=0$$

. . . .

for $i=n_k+m_k-1$

$$\alpha_0 c^{(1)}\big(\xi^{n_k+m_k-1}P_k^{(1)}\big)+\cdots+\alpha_{m_k-1}c^{(1)}\big(\xi^{n_k+2m_k-2}P_k^{(1)}\big)+c^{(1)}\big(\xi^{n_k+2m_k-1}P_k^{(1)}\big)$$
$$+\alpha_0' c^{(1)}\big(\xi^{n_k+m_k-1}P_{k-1}^{(1)}\big)+\cdots+\alpha_{m_k-1}' c^{(1)}\big(\xi^{n_k+2m_k-2}P_{k-1}^{(1)}\big)=0.$$

In the case of breakdown these relations obviously reduce to those of the MRZ.

Let us study the case $n_k - m_k < 0$. We now have $n_k + m_k$ equations, with $n_k + m_k < 2m_k$, for determining the $2m_k$ unknown coefficients. Thus we shall now take for r_k a polynomial of degree at most $n_k - 1$

$$r_k(\xi) = \alpha'_0 + \cdots + \alpha'_{n_k-1}\xi^{n_k-1}.$$

The orthogonality conditions of $P_{k+1}^{(1)}$ give

for $i = 0$

$$\alpha_{n_k} c^{(1)}(\xi^{n_k} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{m_k-1} P_k^{(1)}) + c^{(1)}(\xi^{m_k} P_k^{(1)}) + \alpha'_{n_k-1} c^{(1)}(\xi^{n_k-1} P_{k-1}^{(1)}) = 0$$

. . . .

for $i = n_k - 1$

$$\alpha_1 c^{(1)}(\xi^{n_k} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{n_k+m_k-2} P_k^{(1)}) + c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) + \alpha'_0 c^{(1)}(\xi^{n_k-1} P_{k-1}^{(1)}) + \cdots + \alpha'_{n_k-1} c^{(1)}(\xi^{2n_k-2} P_{k-1}^{(1)}) = 0$$

for $i = n_k$

$$\alpha_0 c^{(1)}(\xi^{n_k} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) + c^{(1)}(\xi^{n_k+m_k} P_k^{(1)}) + \alpha'_0 c^{(1)}(\xi^{n_k} P_{k-1}^{(1)}) + \cdots + \alpha'_{n_k-1} c^{(1)}(\xi^{2n_k-1} P_{k-1}^{(1)}) = 0$$

. . . .

for $i = n_k + m_k - 1$

$$\alpha_0 c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{n_k+2m_k-2} P_k^{(1)}) + c^{(1)}(\xi^{n_k+2m_k-1} P_k^{(1)}) + \alpha'_0 c^{(1)}(\xi^{n_k+m_k-1} P_{k-1}^{(1)}) + \cdots + \alpha'_{n_k-1} c^{(1)}(\xi^{2n_k+m_k-2} P_{k-1}^{(1)}) = 0.$$

In the case of breakdown we again recover the MRZ.

If the systems giving $P_{k+1}^{(1)}$ are singular then $P_{k+1}^{(1)}$ does not exist and the value of m_k has to be increased until a regular polynomial $P_{k+1}^{(1)}$ has been found. Once $P_{k+1}^{(1)}$ has been determined by this method its successor or its predecessor must be computed in order to be able to apply it again. If $c^{(1)}(\xi^i P_k^{(1)}) = 0$ for $i = 0, \ldots, n_k + m_k - 2$ and $c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) \neq 0$ then $P_k^{(1)}$ is the predecessor of $P_{k+1}^{(1)}$ (which means that the regular polynomials of degrees $n_k + 1, \ldots, n_{k+1} - 1$ do not exist) and the GMRZ can be re-applied from $P_k^{(1)}$ and $P_{k+1}^{(1)}$. If this is not the case, the predecessor of $P_{k+1}^{(1)}$ has a degree between $n_k + 1$ and $n_{k+1} - 1$. However, due to the conditions producing the near-breakdown, it will be badly computed. Thus it will be better, from the numerical point of view, to find its successor. First we have to determine m_k such that

$$c^{(1)}(\xi^i P_{k+1}^{(1)}) = 0 \quad \text{for } i = 0, \ldots, n_{k+1} + m_{k+1} - 2$$
$$\neq 0 \quad \text{for } i = n_{k+1} + m_{k+1} - 1$$

and then $P_{k+2}^{(1)}$ has to be computed by the GMRZ from $P_k^{(1)}$ and $P_{k-1}^{(1)}$. The following polynomials could then be obtained by the GMRZ from $P_{k+1}^{(1)}$ and $P_{k+2}^{(1)}$

which are successive regular polynomials as required by the process. Of course, due to rounding errors, m_{k+1} will be in general equal to 1 since the quantity $|c^{(1)}(\xi^{n_{k+1}}P_{k+1}^{(1)})|$ will not be equal to zero exactly.

4.2. THE BSMRZ

Let us now try to generalize to the near-breakdown the relations obtained for $P_{k+1}^{(1)}$ in the two variants of the MRZ.

Since the BMRZ was easier to use we shall first try to generalize it. For that purpose, $P_{k+1}^{(1)}$ shall be expressed as

$$P_{k+1}^{(1)}(\xi) = A_{k+1}P_{k+1}(\xi) + q_k(\xi)P_k^{(1)}(\xi)$$

where q_k is a polynomial of degree $d_k \leqslant m_k$. Thus we must determine $d_k + 2$ unknown coefficients.

Since $c(\xi^{i+1}P_{k+1}) = 0$ for $i = 0, \ldots, n_k + m_k - 2$ and $c^{(1)}(\xi^i q_k P_k^{(1)}) = 0$ for $i = 0, \ldots, n_k - d_k - 1$, the orthogonality conditions of $P_{k+1}^{(1)}$ are satisfied for $i = 0, \ldots, n_k - d_k - 1$. Writing the missing orthogonality conditions and imposing that $P_{k+1}^{(1)}$ is monic leads to $m_k + d_k + 1$ relations which is only possible if $m_k = 1$.

If we try to replace the constant A_{k+1} by a polynomial we never obtain a number of equations equal to the number of unknown coefficients and thus the BMRZ cannot be generalized to treat a near-breakdown.

Thus let us generalize the SMRZ. The method will be called the BSMRZ where B stands for *block*. We shall look for $P_{k+1}^{(1)}$ under the form

$$P_{k+1}^{(1)}(\xi) = q_k(\xi)P_k^{(1)}(\xi) + t_k(\xi)P_k(\xi)$$

where q_k is a monic polynomial of degree m_k and t_k polynomial of degree $m_k - 1$ at most. Thus we have $2m_k$ unknown coefficients and

$$c^{(1)}(\xi^i P_{k+1}^{(1)}) = c^{(1)}(\xi^i q_k P_k^{(1)}) + c(\xi^{i+1} t_k P_k).$$

Since $c^{(1)}(\xi^i q_k P_k^{(1)}) = c(\xi^{i+1} t_k P_k) = 0$ for $i = 0, \ldots, n_k - m_k - 1$ the orthogonality conditions of $P_{k+1}^{(1)}$ are satisfied for the same indexes. Writing these conditions for $i = n_k - m_k, \ldots, n_k + m_k - 1$ gives $2m_k$ relations. Obviously we must assume that $n_k - m_k \geqslant 0$. The case $n_k - m_k < 0$ will be treated below.

We set

$$q_k(\xi) = \alpha_0 + \cdots + \alpha_{m_k-1}\xi^{m_k-1} + \xi^{m_k}$$

$$t_k(\xi) = \alpha'_0 + \cdots + \alpha'_{m_k-1}\xi^{m_k-1}$$

and thus

$$c^{(1)}(\xi^i P_{k+1}^{(1)}) = \alpha_0 c^{(1)}(\xi^i P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{i+m_k-1} P_k^{(1)}) + c^{(1)}(\xi^{i+m_k} P_k^{(1)})$$
$$+ \alpha'_0 c(\xi^{i+1} P_k) + \cdots + \alpha'_{m_k-1} c(\xi^{i+m_k} P_k).$$

The orthogonality conditions of $P_{k+1}^{(1)}$ give

for $i = n_k - m_k$

$$\alpha'_{m_k-1} c(\xi^{n_k} P_k) = -c^{(1)}(\xi^{n_k} P_k^{(1)})$$

for $i = n_k - m_k + 1$

$$\alpha_{m_k-1} c^{(1)}(\xi^{n_k} P_k^{(1)}) + \alpha'_{m_k-2} c(\xi^{n_k} P_k) + \alpha'_{m_k-1} c(\xi^{n_k+1} P_k) = -c^{(1)}(\xi^{n_k+1} P_k^{(1)})$$

....

for $i = n_k - 1$

$$\alpha_1 c^{(1)}(\xi^{n_k} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{n_k+m_k-2} P_k^{(1)}) + \alpha'_0 c(\xi^{n_k} P_k) + \cdots$$
$$+ \alpha'_{m_k-1} c(\xi^{n_k+m_k-1} P_k) = -c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)})$$

for $i = n_k$

$$\alpha_0 c^{(1)}(\xi^{n_k} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) + \alpha'_0 c(\xi^{n_k+1} P_k) + \cdots$$
$$+ \alpha'_{m_k-1} c(\xi^{n_k+m_k} P_k) = -c^{(1)}(\xi^{n_k+m_k} P_k^{(1)})$$

....

for $i = n_k + m_k - 1$

$$\alpha_0 c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{n_k+2m_k-2} P_k^{(1)})$$
$$+ \alpha'_0 c(\xi^{n_k+m_k} P_k) + \cdots + \alpha'_{m_k-1} c(\xi^{n_k+2m_k-1} P_k) = -c^{(1)}(\xi^{n_k+2m_k-1} P_k^{(1)}).$$

In the case of breakdown we recover the SMRZ. If $c(\xi^{n_k} P_k) = 0$ this system is singular and, as in the SMRZ, $P_{k+1}^{(1)}$ cannot be obtained as a combination of $P_k^{(1)}$ and P_k. In this case it is mandatory to use the GMRZ. However this case will seldom occur and most of the time $|c(\xi^{n_k} P_k)|$ will not be equal to zero but small.

Let us now consider the case $n_k - m_k < 0$. We shall have $n_k + m_k < 2m_k$ equations for computing the $2m_k$ unknowns. Thus we shall now take for t_k a polynomial of degree at most $n_k - 1$

$$t_k(\xi) = \alpha'_0 + \cdots + \alpha'_{n_k-1} \xi^{n_k-1}.$$

We have

$$c^{(1)}(\xi^i P_{k+1}^{(1)}) = \alpha_0 c^{(1)}(\xi^i P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{i+m_k-1} P_k^{(1)}) + c^{(1)}(\xi^{i+m_k} P_k^{(1)})$$
$$+ \alpha'_0 c(\xi^{i+1} P_k) + \cdots + \alpha'_{n_k-1} c(\xi^{i+n_k} P_k)$$

and the orthogonality conditions of $P_{k+1}^{(1)}$ give for $i=0$

$$\alpha_{n_k} c^{(1)}(\xi^{n_k} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{m_k-1} P_k^{(1)}) + c^{(1)}(\xi^{m_k} P_k^{(1)}) + \alpha'_{n_k-1} c(\xi^{n_k} P_k) = 0$$

. . . .

for $i = n_k - 1$

$$\alpha_1 c^{(1)}(\xi^{n_k} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{n_k+m_k-2} P_k^{(1)}) + c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) + \alpha'_0 c(\xi^{n_k} P_k) + \cdots + \alpha'_{n_k-1} c(\xi^{2n_k-1} P_k) = 0$$

for $i = n_k$

$$\alpha_0 c^{(1)}(\xi^{n_k} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) + c^{(1)}(\xi^{n_k+m_k} P_k^{(1)}) + \alpha'_0 c(\xi^{n_k+1} P_k) + \cdots + \alpha'_{n_k-1} c(\xi^{2n_k} P_k) = 0$$

. . . .

for $i = n_k + m_k - 1$

$$\alpha_0 c^{(1)}(\xi^{n_k+m_k-1} P_k^{(1)}) + \cdots + \alpha_{m_k-1} c^{(1)}(\xi^{n_k+2m_k-2} P_k^{(1)}) + c^{(1)}(\xi^{n_k+2m_k-1} P_k^{(1)}) + \alpha'_0 c(\xi^{n_k+m_k} P_k) + \cdots + \alpha'_{n_k-1} c(\xi^{2n_k+m_k-1} P_k) = 0.$$

In the case of breakdown we have $\alpha'_1 = \cdots = \alpha'_{n_k-1}$ and the SMRZ is recovered. If the systems giving the coefficients are singular, it means that $P_{k+1}^{(1)}$ does not exist and the value of m_k has to be increased until a regular polynomial $P_{k+1}^{(1)}$ has been obtained.

5. Programming the algorithms

Let us now analyze the coding of the algorithms. It was done such as to minimize the storage and the number of vector operations (scalar products and matrix by vector multiplications). Since the logical design of the algorithms are quite similar, it is sufficient to give only two of them in a pseudo-code, namely the MRZ and the BSMRZ. They are as follows

Algorithm MRZ (A, b, x_0, y)

1. **Initializations:**
   $z_{-1} \leftarrow 0$
   $r_0 \leftarrow Ax_0 - b$
   $s_0 = z_0 = r_0$
   $n_0 \leftarrow 0$
2. **For** $k = 0, 1, 2, \ldots$ **until** convergence **do:**
   If $n_k = n$ **then**
   solution not obtained after n iterations.
   stop.
   end if

$s_1 \leftarrow As_0$
If $(y, s_1) = 0$ **and** $n_k = n - 1$ **then**
incurable breakdown.
stop.
end if
$m_k \leftarrow 1$
3. **While** $(y, s_{m_k}) = 0$ **and** $m_k < n - n_k$ **do:**
$m_k \leftarrow m_k + 1$
$s_{m_k} \leftarrow As_{m_k - 1}$
end while
If $m_k = n - n_k$ **and** $(y, s_{m_k}) = 0$ **then**
incurable breakdown.
stop.
end if
$d_0 \leftarrow (y, r_k)$
$b_0 \leftarrow (y, s_{m_k})$
$\beta_{m_k - 1} \leftarrow d_0/b_0$
4. **For** $i = 1, \ldots, m_k$ **do:**
$y \leftarrow A^{\mathrm{T}}\}y$
$b_i \leftarrow (y, s_{m_k})$
If $i \neq m_k$ **then**
$d_i \leftarrow (y, r_k)$
compute $\beta_{m_k - i - 1}$
end if
If $k \neq 0$ and $i > m_{k-1}$ **then**
$p_i \leftarrow (y, z_{k-1})$
end if
end for
5. compute $x_{k+1} = x_k - w_k(A)z_k$
compute $r_{k+1} = r_k - Aw_k(A)z_k$
If $r_{k+1} = 0$ **then**
solution obtained.
stop.
end if
6. $n_{k+1} \leftarrow n_k + m_k$
If $k = 0$ **then**
$C_1 \leftarrow 0$
$p_0 \leftarrow 0$
else
$C_{k+1} \leftarrow b_0/p_0$
end if
7. **For** $i = 1, \ldots, m_k$ **do:**
If $k = 0$ **then**

$p_i \leftarrow 0$
end if
compute $\alpha_{m_k - i}$
end for
8. **For** $i = 0, \ldots, m_k$ **do:**
$p_i \leftarrow b_i$
end for
$s_0 \leftarrow z_{k+1} = q_k(A) z_k - C_{k+1} z_{k-1}$
end for

Algorithm BSMRZ $(A, b, x_0, y, \varepsilon)$

1. **Initializations:**
$r_0 \leftarrow Ax_0 - b$
$s_0 = z_0 = r_0$
$u_0 = r_0$
$n_0 \leftarrow 0$
2. **For** $k = 0, 1, 2, \ldots$ **until** convergence **do:**
If $n_k = n$ **then**
solution not obtained after n iterations.
stop.
end if
$d_0 \leftarrow (y, r_k)$
If $d_0 = 0$ **then**
impossible to use the BSMRZ.
stop.
end if
$s_1 \leftarrow As_0$
$c_0 \leftarrow (y, s_1)$
If $|c_0| \leqslant \varepsilon$ **and** $n_k = n - 1$ **then**
incurable near-breakdown.
stop.
end if
$m_k \leftarrow 1$
3. **While** $|c_{m_k - 1}| \leqslant \varepsilon$ **and** $m_k < n - n_k$ **do:**
$m_k \leftarrow m_k + 1$
$y \leftarrow A^{\mathrm{T}} y$
$c_{m_k - 1} \leftarrow (y, s_1)$
$d_{m_k - 1} \leftarrow (y, r_k)$
$s_{m_k} \leftarrow As_{m_k - 1}$
end while
If $m_k = n - n_k$ **and** $c_{m_k - 1} \leqslant \varepsilon$ **then**
incurable near-breakdown.

```
        stop.
      end if
      y ← A^T y
      ŷ ← y
      c_{m_k} ← (y, s_1)
      If n_k ≠ 0 then d_{m_k} ← (y, r_k)
      ŷ ← y
4.    If m_k ≠ 1 then
        For i = 1, ..., m_k − 1 do:
          ŷ ← A^T ŷ
          c_{m_k+i} ← (ŷ, s_1)
          If i < n_k then d_{m_k+i} ← (ŷ, r_k)
          If i ⩽ n_k then u_i ← A u_{i−1}
        end for
      end if
5.    Repeat
        If m_k ⩽ n_k then
          compute β_i, (i = 0, ..., m_k − 1)
          compute β'_j, (j = 0, ..., m_k − 2)
          compute α_i, (i = 0, ..., m_k − 1)
          compute α'_j, (j = 0, ..., m_k − 1)
        else
          compute β_i, (i = 0, ..., m_k − 1)
          compute β'_j, (j = 0, ..., n_k − 1)
          compute α_i, (i = 0, ..., m_k − 1)
          compute α'_j, (j = 0, ..., n_k − 1)
        end if
6.      If singular system then
          m_k ← m_k + 1
          If m_k + n_k = n + 1 then
            incurable near-breakdown.
            stop.
          end if
          y ← A^T y
          For i = 2 downto 1 do:
            ŷ ← A^T ŷ
            c_{2m_k−i} ← (ŷ, s_1)
            d_{2m_k−i} ← (ŷ, r_k)
          end for
          s_{m_k} ← A s_{m_k−1}
          If m_k − 1 ⩽ n_k then u_{m_k−1} ← A u_{m_k−2}
        end if
      until not singular system.
```

7. compute $x_{k+1} = x_k - w_k(A)z_k - v_k(A)r_k$
   compute $r_{k+1} = r_k - Aw_k(A)z_k - Av_k(A)r_k$
   If $r_{k+1} = 0$ **then**
     solution obtained.
     stop.
   end if
8. $n_{k+1} \leftarrow n_k + m_k$
   $s_0 \leftarrow z_{k+1} = q_k(A)z_k + t_k(A)r_k$
   $u_0 \leftarrow r_{k+1}$
end for

This coding needs the storage of $m+6$ vectors where $m = max_k m_k$. Thus, in the case where no breakdown occurs, $m = 1$ and 7 auxiliary vectors are used which is exactly the number of auxiliary vectors needed in the BIODIR algorithm and shows the optimality of our programming with respect to storage.

The subroutines have been programmed in FORTRAN 77. They can be obtained by electronic mail via *netlib*. The command to be used is

netlib@research.att.com

and the name of the library is NUMERALGO. To obtain the program xxx ask

send xxx from numeralgo

They are the following

MRZ	algorithm MRZ
BMRZ	algorithm BMRZ
SMRZ	algorithm SMRZ
BSMRZ	algorithm BSMRZ

They are given together with the corresponding main programs (same name as the subroutine preceded by an M).

It is possible to compute differently the scalar products needed in the algorithms. Indeed, since $P_k^{(1)}$ has the degree n_k exactly, the relations $c(\xi^i P_{k+1}) = 0$ for $i = n_k, \ldots, n_k + m_k - 1$ can be replaced by $c(\xi^i P_k^{(1)} P_{k+1}) = 0$ for $i = 0, \ldots, m_k - 1$. Thus, in the MRZ, we have

$$c\left(\xi^i P_k^{(1)} P_{k+1}\right) = c\left(\xi^i P_k^{(1)} P_k\right) - \beta_0 c^{(1)}\left(\xi^i P_k^{(1)^2}\right) - \cdots - \beta_{m_{k-1}} c^{(1)}\left(\xi^{i+m_k-1} P_k^{(1)^2}\right).$$

But $c^{(1)}(\xi^i P_k^{(1)^2}) = 0$ for $i = 0, \ldots, m_k - 2$ since $P_k^{(1)}$ is orthogonal to any polynomial of degree at most $n_k + m_k - 2$ and we obtain

$$\begin{cases} \beta_{m_k-1} c^{(1)}\left(\xi^{m_k-1} P_k^{(1)^2}\right) = c\left(P_k^{(1)} P_k\right) \\ \beta_{m_k-2} c^{(1)}\left(\xi^{m_k-1} P_k^{(1)^2}\right) + \beta_{m_k-1} c^{(1)}\left(\xi^{m_k} P_k^{(1)^2}\right) = c\left(\xi P_k^{(1)} P_k\right) \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \beta_0 c^{(1)}\left(\xi^{m_k-1} P_k^{(1)^2}\right) + \cdots + \beta_{m_k-1} c^{(1)}\left(\xi^{2m_k-2} P_k^{(1)^2}\right) = c\left(\xi^{m_k-1} P_k^{(1)} P_k\right). \end{cases}$$

Thus we need to compute $c^{(1)}(\xi^i P_k^{(1)^2})$ for $i \geqslant m_k - 1$ and $c^{(1)}(\xi^i P_k^{(1)} P_k)$ for $i \geqslant 0$. We have

$$c^{(1)}\left(\xi^i P_k^{(1)^2}\right) = \left(y, A^{i+1} P_k^{(1)}(A) P_k^{(1)}(A) r_0\right)$$
$$= \left(P_k^{(1)}(A)^* y, A^{i+1} P_k^{(1)}(A) r_0\right).$$

If we set $z_k = P_k^{(1)}(A) r_0$ and $\tilde{z}_k = P_k^{(1)}(A)^* y = \overline{P}_k^{(1)}(A^*) y$ then

$$c^{(1)}\left(\xi^i P_k^{(1)^2}\right) = \left(\tilde{z}_k, A^{i+1} z_k\right).$$

Similarly

$$c\left(\xi^i P_k^{(1)} P_k\right) = \left(y, A^i P_k^{(1)}(A) P_k(A) r_0\right)$$
$$= \left(P_k^{(1)}(A)^* y, A^i P_k(A) r_0\right)$$
$$= \left(\tilde{z}_k, A^i r_k\right).$$

In the three-term recurrence relationship of the polynomials $P_k^{(1)}$, the relations $c^{(1)}(\xi^i P_{k+1}^{(1)}) = 0$ for $i = n_k, \dots n_k + m_k - 1$ can be replaced by $c^{(1)}(\xi^i P_k^{(1)} P_{k+1}^{(1)}) = 0$ for $i = 0, \dots, m_k - 1$ since $P_k^{(1)}$ has the exact degree n_k. Moreover $c^{(1)}(\xi^{n_k - 1} P_{k+1}^{(1)}) = 0$ can be replaced by $c^{(1)}(\xi^{m_{k-1}-1} P_{k-1}^{(1)} P_{k+1}^{(1)}) = 0$ since $P_{k-1}^{(1)}$ is exactly of degree n_{k-1} and $n_{k-1} + m_{k-1} = n_k$. Thus we have

$$\begin{aligned} c^{(1)}\left(\xi^{n_k - 1} P_{k+1}^{(1)}\right) &= c^{(1)}\left(\xi^{m_{k-1}-1} P_{k-1}^{(1)} P_{k+1}^{(1)}\right) = 0 \\ &= \alpha_0 c^{(1)}\left(\xi^{m_{k-1}-1} P_{k-1}^{(1)} P_k^{(1)}\right) + \cdots + \alpha_{m_k - 1} c^{(1)}\left(\xi^{m_{k-1}+m_k-2} P_{k-1}^{(1)} P_k^{(1)}\right) \\ &\quad + c^{(1)}\left(\xi^{m_{k-1}+m_k-1} P_{k-1}^{(1)} P_k^{(1)}\right) - C_{k+1} c^{(1)}\left(\xi^{m_{k-1}-1} P_{k-1}^{(1)^2}\right). \end{aligned}$$

Thanks to the orthogonality of $P_k^{(1)}$ we have

$$C_{k+1} c^{(1)}\left(\xi^{m_{k-1}-1} P_{k-1}^{(1)^2}\right) = c^{(1)}\left(\xi^{m_{k-1}+m_k-1} P_{k-1}^{(1)} P_k^{(1)}\right)$$

and

$$\begin{aligned} c^{(1)}\left(\xi^{n_k} P_{k+1}^{(1)}\right) &= c^{(1)}\left(P_k^{(1)} P_{k+1}^{(1)}\right) = 0 \\ &= \alpha_0 c^{(1)}\left(P_k^{(1)^2}\right) + \cdots + \alpha_{m_k - 1} c^{(1)}\left(\xi^{m_k - 1} P_k^{(1)^2}\right) \\ &\quad + c^{(1)}\left(\xi^{m_k} P_k^{(1)^2}\right) - C_{k+1} c^{(1)}\left(P_k^{(1)} P_{k-1}^{(1)}\right). \end{aligned}$$

Finally, since $c^{(1)}(\xi^i P_k^{(1)} P_{k-1}^{(1)}) = 0$ for $i + n_{k-1} \leqslant n_k + m_k - 2$, that is for $i \leqslant m_{k-1} + m_k - 2$ with $m_{k-1} \geqslant 1$, we obtain

$$\begin{cases} \alpha_{m_k - 1} c^{(1)}\left(\xi^{m_k - 1} P_k^{(1)^2}\right) + c^{(1)}\left(\xi^{m_k} P_k^{(1)^2}\right) = 0 \\ \alpha_{m_k - 2} c^{(1)}\left(\xi^{m_k - 1} P_k^{(1)^2}\right) + \alpha_{m_k - 1} c^{(1)}\left(\xi^{m_k} P_k^{(1)^2}\right) + c^{(1)}\left(\xi^{m_k + 1} P_k^{(1)^2}\right) = 0 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ \alpha_0 c^{(1)}\left(\xi^{m_k - 1} P_k^{(1)^2}\right) + \cdots + \alpha_{m_k - 1} c^{(1)}\left(\xi^{2m_k - 2} P_k^{(1)^2}\right) + c^{(1)}\left(\xi^{2m_k - 1} P_k^{(1)^2}\right) = 0. \end{cases}$$

Thus, for obtaining C_{k+1}, we have to compute $c^{(1)}(\xi^{m_{k-1}+m_k-1} P_{k-1}^{(1)} P_k^{(1)})$. But $\xi^{m_{k-1}} P_{k-1}^{(1)}$ is monic of degree $n_{k-1} + m_{k-1} = n_k$, $\xi^{m_{k-1}+m_k-1} P_{k-1}^{(1)}$ is monic of degree $n_k + m_k - 1$ and $P_k^{(1)}$ is orthogonal to any polynomial of degree strictly less than $n_k + m_k - 1$. It follows that

$$c^{(1)}\left(\xi^{m_{k-1}+m_k-1}P_{k-1}^{(1)}P_k^{(1)}\right) = c^{(1)}\left(\xi^{m_k-1}P_k^{(1)^2}\right)$$

since $P_{k-1}^{(1)}$ and $P_k^{(1)}$ are monic and finally

$$C_{k+1}c^{(1)}\left(\xi^{m_{k-1}-1}P_{k-1}^{(1)^2}\right) = c^{(1)}\left(\xi^{m_k-1}P_k^{(1)^2}\right).$$

Moreover, by the recurrence relationship of the polynomials $P_k^{(1)}$

$$\tilde{z}_{k+1} = q_k(A)^* \tilde{z}_k - \bar{C}_{k+1}\tilde{z}_{k-1} = \bar{q}_k(A^*)\tilde{z}_k - \bar{C}_{k+1}\tilde{z}_{k-1}.$$

Computing the coefficients in that way, leads to a method similar to the BIODIR method given by Gutknecht [8].

Obviously, for testing a breakdown, a strict equality to zero of a scalar product can never be achieved because of the rounding errors. Thus, in our programs, this condition is replaced by the absolute value of the scalar product less than a given ε.

6. Numerical results

Let us consider the system [1]

$$\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ n \end{pmatrix} = \begin{pmatrix} -n \\ 1 \\ 2 \\ \vdots \\ n-1 \end{pmatrix}.$$

With the subroutine MRZ we obtain the following results for $n = 12$, $x_0 = 0$, $y = (1, \ldots, 1)^T$ and $\varepsilon = 10^{-1}$

k	1	2	3	4	5	6	7	8	9	10
n_k	1	2	3	6	7	8	9	10	11	12
$\lVert r_k \rVert$	15.0	18.3	37.5	41.1	41.1	945.3	948.8	37.7	18.3	7.0

For $\varepsilon = 10^{-2}, 10^{-3}, 10^{-5}, 10^{-10}, 10^{-11}$, we obtain

k	1	2	3	4	5	6	7	8
n_k	1	2	3	4	9	10	11	12
$\lVert r_k \rVert$	15.0	18.3	37.5	58.2	58.2	37.6	18.2	$3.2 \cdot 10^{-9}$

For $\varepsilon = 10^{-12}$, we have

k	1	2	3	4	5	6	7	8	9
n_k	1	2	3	4	8	9	10	11	12
$\lVert r_k \rVert$	15.0	18.3	37.5	58.2	$2.5 \cdot 10^{41}$	$2.9 \cdot 10^{25}$	$9.8 \cdot 10^{24}$	$8.1 \cdot 10^{25}$	$2.8 \cdot 10^{26}$

For $\varepsilon = 10^{-13}$, we obtain

k	1	2	3	4	5	6	7	8	9	10	11
n_k	1	2	3	4	6	7	8	9	10	11	12
$\|r_k\|$	15.0	18.3	37.5	58.2	47.7	47.7	$1.0 \cdot 10^{14}$	58.2	42.3	54.3	$8.4 \cdot 10^4$

Finally for $\varepsilon = 10^{-14}$, 10^{-15} and 10^{-16} we find

k	1	2	3	4	5	6	7	8	9	10	11	12
n_k	1	2	3	4	5	6	7	8	9	10	11	12
$\|r_k\|$	15.0	18.3	37.5	58.2	63.6	73.6	73.6	63.6	58.2	37.6	18.2	57.4

Thus the results are quite sensitive to the value of ε which is not surprising since this value controls the correct detection of the jump between the dimensions 4 and 9. When this jump is correctly determined then the exact solution of the system is obtained. Let us mentioned that the computation was performed on a personal computer working with 16 decimal digits in double precision.

Let us consider again the same system and give the value of the norm of the last residual obtained by the various algorithms. When no number is indicated, it means that the solution has not been obtained (which is due to a division by zero in the algorithm because the required supplementary assumption is not satisfied). In the BSMRZ, if the solution was not obtained after n iterations, we let them continue; in that case, the integer placed into parenthesis indicates the number of iterations performed. The value of ε_1 used in the BSMRZ corresponds to the threshold for testing the pivots in Gaussian elimination for solving the auxiliary systems. When a pivot has an absolute value less than ε_1, the value of m_k is increased by 1.

We obtained the following results with $x_0 = 0$ and $y = (1, \ldots, 1)^T$

n	MRZ	BMRZ	SMRZ	BSMRZ	BSMRZ	BSMRZ
	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-8}$, $\varepsilon_1 = 10^{-14}$	$\varepsilon = 10^{-1}$, $\varepsilon_1 = 10^{-11}$	$\varepsilon = 1$, $\varepsilon_1 = 10^{-11}$
4	$2.74 \cdot 10^{-15}$	$1.58 \cdot 10^{-15}$	$1.46 \cdot 10^{-15}$	$1.83 \cdot 10^{-15}$	$1.83 \cdot 10^{-15}$	$1.83 \cdot 10^{-15}$
5	$7.20 \cdot 10^{-15}$	$6.21 \cdot 10^{-14}$	$1.62 \cdot 10^{-13}$	$1.65 \cdot 10^{-13}$	$1.65 \cdot 10^{-13}$	$1.65 \cdot 10^{-13}$
6	$1.33 \cdot 10^{-11}$	$5.39 \cdot 10^{-14}$	$2.63 \cdot 10^{-13}$	$1.47 \cdot 10^{-13}$	$1.47 \cdot 10^{-13}$	$3.72 \cdot 10^{-14}$
7	$5.49 \cdot 10^{-13}$			$8.34 \cdot 10^{-14}$(11)	$2.13 \cdot 10^{-13}$	$3.45 \cdot 10^{-13}$
8	$6.53 \cdot 10^{-12}$			$4.59 \cdot 10^{-14}$(12)	$2.09 \cdot 10^{-12}$	$1.96 \cdot 10^{-12}$
9	$4.23 \cdot 10^{-11}$			$1.72 \cdot 10^{-14}$(13)	$2.48 \cdot 10^{-12}$	$2.23 \cdot 10^{-12}$
10	$5.09 \cdot 10^{-11}$			$5.07 \cdot 10^{-14}$(14)	$1.63 \cdot 10^{-12}$	$3.29 \cdot 10^{-12}$
11	$1.10 \cdot 10^{-11}$				$5.48 \cdot 10^{-13}$(24)	$3.86 \cdot 10^{-12}$
12	$3.33 \cdot 10^{-11}$					$1.68 \cdot 10^{-12}$

With $y = r_0$ we found

n	MRZ	BMRZ	SMRZ	BSMRZ	BSMRZ	BSMRZ
	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-8}$	$\varepsilon = 10^{-8}$ $\varepsilon_1 = 10^{-14}$	$\varepsilon = 10^{-1}$ $\varepsilon_1 = 10^{-11}$	$\varepsilon = 1$ $\varepsilon_1 = 10^{-11}$
4	0.0					
5	$1.06 \cdot 10^{-10}$	$3.39 \cdot 10^{-13}$	$8.12 \cdot 10^{-13}$	$4.27 \cdot 10^{-13}$	$2.56 \cdot 10^{-13}$	$2.56 \cdot 10^{-13}$
6	$2.32 \cdot 10^{-8}$	$1.53 \cdot 10^{-10}$	$1.90 \cdot 10^{-10}$	$1.09 \cdot 10^{-10}$	$1.09 \cdot 10^{-10}$	$2.22 \cdot 10^{-13}$
7	$3.02 \cdot 10^{-10}$	$2.62 \cdot 10^{-12}$	$3.54 \cdot 10^{-12}$	$4.00 \cdot 10^{-12}$	$2.08 \cdot 10^{-12}$	$2.08 \cdot 10^{-12}$
8	$2.04 \cdot 10^{-11}$			$3.39 \cdot 10^{-12}$(13)	$3.66 \cdot 10^{-13}$(12)	$1.21 \cdot 10^{-12}$
9	$4.20 \cdot 10^{-11}$			$2.77 \cdot 10^{-12}$(14)		$1.87 \cdot 10^{-12}$
10	$4.57 \cdot 10^{-10}$			$4.72 \cdot 10^{-12}$(15)		$1.74 \cdot 10^{-12}$
11	$5.76 \cdot 10^{-10}$			$4.63 \cdot 10^{-12}$(16)		$5.88 \cdot 10^{-12}$
12	$1.80 \cdot 10^{-9}$			$2.11 \cdot 10^{-12}$(17)	$1.21 \cdot 10^{-10}$	$3.73 \cdot 10^{-12}$

These results seem to show that the BMRZ and the SMRZ are more stable than the MRZ. The BSMRZ gives better results than the MRZ but is quite sensitive to the choice of ε and ε_1. Thus more experiments are needed in order to fully understand the numerical behaviour of these algorithms. A theoretical study of their stability is also necessary and procedures (such as reorthogonalization and preconditioning) for improving their numerical performances have to be tried. Gaussian elimination in the BSMRZ has also to be replaced by a better method.

We intend to come back to these questions in the future.

References

[1] D.L. Boley, S. Elhay, G.H. Golub and M.H. Gutknecht, Nonsymmetric Lanczos and finding orthogonal polynomials associated with indefinite weights, Numer. Algorithms 1 (1991) 21–44.

[2] C. Brezinski, *Padé-type Approximation and General Orthogonal Polynomials.*, ISNM Vol. 50, (Birkhäuser, Basel, 1980).

[3] C. Brezinski, M. Redivo Zaglia and H. Sadok, A breakdown-free Lanczos type algorithm for solving linear systems, Numer. Math., to appear.

[4] C. Brezinski and H. Sadok, Lanczos type methods for systems of linear equations, submitted.

[5] C. Brezinski and H. Sadok, Avoiding breakdown in the CGS algorithm, Numer. Algorithms 1 (1991) 199–206.

[6] A. Draux, *Polynômes Orthogonaux Formels. Applications.*, LNM 974 (Springer-Verlag, Berlin, 1983).

[7] R. Fletcher, Conjugate gradient methods for indefinite systems, in: *Numerical Analysis*, ed. G.A. Watson, LNM 506 (Springer-Verlag, Berlin, 1976) pp. 73–89.

[8] M.H. Gutknecht, A completed theory of the unsymmetric Lanczos process and related algorithms, Parts I, II, SIAM J. Matrix Anal. Appl., to appear.

[9] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems, SIAM J. Sci. Stat. Comp. 10 (1989) 36–52.