SIAM J. MATRIX ANAL. APPL.
Vol. 15, No. 3, pp. 922–937, July 1994

014

THE REVERSE BORDERING METHOD*

C. BREZINSKI†, M. MORANDI CECCHI‡, AND M. REDIVO-ZAGLIA§

Abstract. The bordering method allows recursive computation of the solution of a system of linear equations by adding one new row and one new column at each step of the procedure. When some of the intermediate systems are nearly singular, it is possible, by the block bordering method, to add several new rows and columns simultaneously. However, in that case, the solutions of some of the intermediate systems are not computed. The reverse bordering method allows computation of the solutions of these systems afterwards. Such a procedure has many applications in numerical analysis, that include orthogonal polynomials, Padé approximation, and the progressive forms of extrapolation processes.

Key words. linear equations, bordering methods, extrapolation, Padé approximants, orthogonal polynomials

AMS subject classifications. 65F05, 65B05

1. Introduction.

The bordering method is a recursive method for computing the solution of a system of linear equations. It consists, at each step, of adding one new row and one new column to the previous matrix and using the previous solution to compute the new one. This method can only be used if some quantity is different from zero at each step. If, at some steps, this quantity is zero (or nearly zero), then it is possible to add several new rows and columns to the matrix simultaneously. However, if this situation occurs, the solutions of the intermediate systems that have been skipped are not computed. This is a drawback of the method since, in some applications, the solutions of all the intermediate systems must be known (if nonsingular). In this paper we propose the reverse bordering method for avoiding this case. The procedure is, after jumping over the near-singular intermediate systems and computing the solution of the first nonnear-singular system, to go back by decreasing the dimension of the matrix (that is, by deleting the last row and the last column) and using the solution of the previous larger system for computing the solutions of the smaller systems that have been skipped. Such a procedure has applications in the recursive computation of orthogonal polynomials, in Padé approximation, and in the implementation of the progressive forms of extrapolation algorithms.

2. Bordering method.

When we must solve a system of linear equations that is obtained by adding one new equation and one new unknown to a given system or, in other words, when the matrix of the system has been bordered by a new row

* Received by the editors March 17, 1992; accepted for publication (in revised form) January 26, 1993.

† Laboratoire d'Analyse Numérique et d'Optimisation, UFR IEEA-M3, Université des Sciences et Technologies de Lille, F-59655 Villeneuve d'Ascq Cedex, France (brezinsk@omega.univ-lille1.fr).

‡ Dipartimento di Matematica Pura ed Applicata, Università degli Studi di Padova, via Belzoni 7, I-35131 Padova, Italy (mcecchi@pdmat1.unipd.it).The work of this author was partially supported by Ministero dell'Università e della Ricerca Scientifica e Tecnologica (Italy) (National Project Analisi Numerica e Matematica Computazionale) and by Consiglio Nazionale delle Ricerche (Italy), Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo, Sottoprogetto Calcolo Scientifico per Grandi Sistemi.

§ Dipartimento di Elettronica e Informatica, Università degli Studi di Padova, via Gradenigo 6/a, I-35131 Padova, Italy (elen@elett1.dei.unipd.it).The work of this author was partially supported by Ministero dell'Università e della Ricerca Scientifica e Tecnologica (Italy) (National Project Analisi Numerica e Matematica Computazionale) and by Consiglio Nazionale delle Ricerche (Italy), Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo, Sottoprogetto Calcolo Scientifico per Grandi Sistemi.

and a new column, the bordering method can be used to save computational time. This method is well known in numerical analysis and permits us to solve a system recursively by using the solution of the previous system (see, Faddeeva [7]). Let us first explain this method.

Let A_k be a regular square matrix of dimension k and d_k a vector of dimension k. Let z_k be the solution of the system

$$A_k z_k = d_k.$$

Now let u_k be a column vector of dimension k, v_k a row vector of dimension k, and a_k a scalar. We consider the bordered matrix A_{k+1} of dimension $k+1$ given by

$$A_{k+1} = \begin{pmatrix} A_k & u_k \\ v_k & a_k \end{pmatrix}.$$

We have

$$A_{k+1}^{-1} = \begin{pmatrix} A_k^{-1} + A_k^{-1} u_k v_k A_k^{-1}/\beta_k & -A_k^{-1} u_k/\beta_k \\ -v_k A_k^{-1}/\beta_k & 1/\beta_k \end{pmatrix}$$

with $\beta_k = a_k - v_k A_k^{-1} u_k$.

Let f_k be a scalar and z_{k+1} be the solution of the bordered system

$$A_{k+1} z_{k+1} = d_{k+1} = \begin{pmatrix} d_k \\ f_k \end{pmatrix}.$$

Then we have

$$z_{k+1} = \begin{pmatrix} z_k \\ 0 \end{pmatrix} + \frac{f_k - v_k z_k}{\beta_k} \begin{pmatrix} -A_k^{-1} u_k \\ 1 \end{pmatrix}.$$

This formula gives the solution of the bordered system in terms of the solution of the previous system.

To avoid computation and storage of A_k^{-1}, we can set $q_k = -A_k^{-1} u_k$ and compute it recursively by the same bordering method. In such a way, we obtain the following variant of the bordering method that needs the storage of A_k instead of that of A_k^{-1} for the original procedure.

Let $q_k^{(i)}$ be the solution of the system

$$A_i q_k^{(i)} = -u_k^{(i)},$$

where $u_k^{(i)}$ is the vector formed by the first i components of u_k. Thus $u_i^{(i)} = u_i$ and $q_i^{(i)} = q_i$ for all i. A_i is the matrix of dimension i formed by the first i rows and columns of A_k.

We have, since A_1 is a number,

$$q_k^{(1)} = -\frac{u_k^{(1)}}{A_1},$$

$$q_k^{(i+1)} = \begin{pmatrix} q_k^{(i)} \\ 0 \end{pmatrix} - \frac{u_{k,i+1} + v_i q_k^{(i)}}{a_i + v_i q_i^{(i)}} \begin{pmatrix} q_i^{(i)} \\ 1 \end{pmatrix}, \qquad i = 1, \ldots, k-1,$$

where $u_{k,i+1}$ is the $(i+1)$th component of u_k.

Then

$$q_k^{(k)} = q_k = -A_k^{-1} u_k,$$

thus allowing us to use the previous formula for computing z_{k+1} without knowledge of A_k^{-1}; [3] and [4] contain the subroutine BORDER performing this variant of the bordering method.

3. Block bordering method. The bordering method can be applied only if $\beta_k \neq 0$ for all k. When this is not the case, we can use a block bordering procedure (see Brezinski, Redivo-Zaglia, and Sadok [5]).

We now assume the following dimensions for the matrices involved in the process

$$\begin{array}{ll} A_k & n_k \times n_k, \\ u_k & n_k \times p_k, \\ v_k & p_k \times n_k, \\ a_k & p_k \times p_k, \end{array}$$

and finally

$$A_{k+1} \qquad n_{k+1} \times n_{k+1}$$

with $n_{k+1} = n_k + p_k$.

We set

$$\beta_k = a_k - v_k\, A_k^{-1}\, u_k$$

and we have

$$A_{k+1}^{-1} = \begin{pmatrix} A_k^{-1} + A_k^{-1} u_k \beta_k^{-1} v_k A_k^{-1} & -A_k^{-1} u_k \beta_k^{-1} \\ -\beta_k^{-1} v_k A_k^{-1} & \beta_k^{-1} \end{pmatrix}.$$

f_k is now a vector with p_k components and we obtain

$$z_{k+1} = \begin{pmatrix} z_k \\ 0 \end{pmatrix} + \begin{pmatrix} -A_k^{-1} u_k \\ I_k \end{pmatrix} \beta_k^{-1} \left(f_k - v_k z_k\right),$$

where I_k is the identity matrix of dimension p_k.

The subroutine BLBORD given in [4] performs this block bordering method.

Remark. We note that the subroutine BLBORD only works if $a_{11} = 1$, which can always be made true. It is also possible to add the instruction A(1,1)=1.0D0/A(1,1) after the instruction Z(1)=D(1)/A(1,1).

Again it is possible to avoid computation and storage of A_k^{-1} by setting $q_k = -A_k^{-1} u_k$ (whose dimension is $n_k \times p_k$) and computing it recursively by the bordering method in the following way.

Let $u_k^{(i)}$ be the $n_i \times p_k$ matrix formed by the first n_i rows of u_k for $i \leq k$, $n_i \leq n_k$. We have $u_i^{(i)} = u_i$. Let $q_k^{(i)}$ be the $n_i \times p_k$ matrix satisfying $A_i q_k^{(i)} = -u_k^{(i)}$ for $i \leq k$. We have $q_i^{(i)} = q_i$.

We set

$$q_k^{(1)} = -A_1^{-1}\, u_k^{(1)}$$

and then we have

$$q_k^{(i+1)} = \begin{pmatrix} q_k^{(i)} \\ 0 \end{pmatrix} - \begin{pmatrix} q_i^{(i)} \\ I_i \end{pmatrix} \beta_i^{-1} \left(u_{k,i+1} + v_i q_k^{(i)} \right), \qquad i = 1, \ldots, k-1$$

with $\beta_i = a_i + v_i q_i^{(i)}$ and $u_{k,i+1}$ the matrix formed by the rows $n_i + 1, \ldots, n_i + p_i$ of u_k.

Instead of using the block bordering method when β_k is zero for some k, it is possible to use a pivoting strategy. If, for some k, $\beta_k = 0$, then the last row of the matrix can be interchanged with the next one and so on until some $\beta_k \neq 0$ has been obtained. Such a procedure is not adapted when the solutions of the intermediate systems must be computed. It can be used if only the last solution is needed and if the solutions of the intermediate systems are not required.

Obviously, the block bordering method can also be used even if the matrix β_k is nonsingular and thus, at each step, an arbitrary number of new rows and columns can be added. In particular, when some of the intermediate systems are almost singular, such a strategy allows us to jump over them and thus improve the numerical stability and precision of the solutions of the subsequent systems. However, in such a case the solutions of the systems that have been skipped have not been computed. The reverse bordering method that we now present allows us to come back afterwards to these systems by deleting rows and columns one by one and obtain their solutions from the solution of the larger system.

4. Reverse bordering method. Let us now look at the possibility of finding A_k^{-1} from A_{k+1}^{-1}.

We write the inverse matrix A_{k+1}^{-1} of dimension n_{k+1} under the form

$$A_{k+1}^{-1} = \begin{array}{cc} \begin{array}{cc} n'_k & p'_k \end{array} & \\ \begin{pmatrix} A'_k & u'_k \\ v'_k & a'_k \end{pmatrix} & \begin{array}{c} n'_k \\ p'_k \end{array} \end{array} .$$

The matrix A_{k+1} will also be partitioned by blocks with the same corresponding dimensions. Thus A_k will be the square matrix of dimension $n'_k = n_{k+1} - p'_k$ obtained by suppressing the last p'_k rows and columns of A_{k+1}.

From the block bordering method we know that

$$\begin{aligned} A'_k &= A_k^{-1} + A_k^{-1} u_k \beta_k^{-1} v_k A_k^{-1} \\ u'_k &= -A_k^{-1} u_k \beta_k^{-1}, \qquad v'_k = -\beta_k^{-1} v_k A_k^{-1}, \\ a'_k &= \beta_k^{-1}, \qquad \beta_k = a_k - v_k\, A_k^{-1}\, u_k. \end{aligned}$$

Because

$$a_k'^{-1} = \beta_k$$

then

$$u'_k\, a_k'^{-1} = -A_k^{-1}\, u_k,$$

and thus

$$u'_k\, a_k'^{-1}\, v'_k = A_k^{-1} u_k \beta_k^{-1} v_k A_k^{-1}.$$

Thus using this in the expression of A'_k gives us

$$A_k^{-1} = A'_k - u'_k \, a'^{-1}_k \, v'_k. \tag{1}$$

This formula, which corresponds to the Schur complement, was already given by Duncan in 1944 [6]. The following relations also hold.

$$\det A_k^{-1} = \det A_{k+1}^{-1} / \det a'_k,$$
$$\det A_{k+1} = \det A_k \cdot \det \beta_k.$$

Moreover, from the Sherman–Morrison formula (see [8] for review), we have

$$A'_k = \left(A_k - u_k \, a_k^{-1} \, v_k\right)^{-1},$$

and

$$A_k = \left(A'_k - u'_k \, a'^{-1}_k \, v'_k\right)^{-1} = A'^{-1}_k + A'^{-1}_k \, u'_k \, \beta'^{-1}_k \, v'_k \, A'^{-1}_k,$$

with

$$\beta'_k = a'_k - v'_k \, A'^{-1}_k \, u'_k.$$

This is another proof of (1).

From these formulæ, we obtain

$$\begin{aligned} A_k &= A'^{-1}_k + u_k \, a_k^{-1} \, v_k = \left(A'_k - u'_k \, a'^{-1}_k \, v'_k\right)^{-1}, \\ A'_k &= A_k^{-1} + u'_k \, a'^{-1}_k \, v'_k = \left(A_k - u_k \, a_k^{-1} \, v_k\right)^{-1}. \end{aligned}$$

Now we want to compute the solution z_k of the previous system

$$A_k z_k = d_k$$

starting from the solution z_{k+1} of the bordered system

$$A_{k+1} z_{k+1} = d_{k+1} = \begin{pmatrix} d_k \\ f_k \end{pmatrix} \begin{matrix} n'_k \\ p'_k \end{matrix}.$$

As previously stated

$$\begin{aligned} z_{k+1} = \begin{pmatrix} z'_k \\ c_k \end{pmatrix} &= \begin{pmatrix} z_k \\ 0 \end{pmatrix} + \begin{pmatrix} -A_k^{-1} u_k \\ I_k \end{pmatrix} \beta_k^{-1} \left(f_k - v_k z_k\right) \\ &= \begin{pmatrix} z_k \\ 0 \end{pmatrix} + \begin{pmatrix} u'_k \\ a'_k \end{pmatrix} \left(f_k - v_k z_k\right). \end{aligned}$$

Thus

$$c_k = a'_k f_k - a'_k v_k z_k,$$

that is,

$$a'^{-1}_k c_k = f_k - v_k z_k \quad \text{or} \quad v_k z_k = f_k - a'^{-1}_k c_k$$

and

$$z'_k = z_k + u'_k \left(f_k - v_k z_k\right) = z_k + u'_k \, a'^{-1}_k \, c_k.$$

Finally, it holds that

$$z_k = z_k' - u_k'\, a_k'^{-1}\, c_k.$$

Another way of finding z_k is as follows. From (1), we have

$$A_k^{-1} d_k = A_k' d_k - u_k'\, a_k'^{-1}\, v_k'\, d_k.$$

But

$$\begin{pmatrix} z_k' \\ c_k \end{pmatrix} = \begin{pmatrix} A_k' & u_k' \\ v_k' & a_k' \end{pmatrix} \begin{pmatrix} d_k \\ f_k \end{pmatrix} = \begin{pmatrix} A_k' d_k + u_k' f_k \\ v_k' d_k + a_k' f_k \end{pmatrix}.$$

Thus

$$A_k' d_k = z_k' - u_k' f_k \quad \text{and} \quad v_k' d_k = c_k - a_k' f_k$$

and we have

$$\begin{aligned} z_k = A_k^{-1} d_k &= z_k' - u_k' f_k - u_k' a_k'^{-1} (c_k - a_k' f_k) \\ &= z_k' - u_k' f_k - u_k'\, a_k'^{-1}\, c_k + u_k'\, a_k'^{-1}\, a_k'\, f_k \\ &= z_k' - u_k' a_k'^{-1} c_k. \end{aligned}$$

5. Variants and particular cases. Instead of bordering the matrix A_k as we did, we can also add the new rows and columns on the top and on the left according to the scheme

$$A_{k+1} = \begin{pmatrix} a_k & v_k \\ u_k & A_k \end{pmatrix}.$$

Thus the inverse matrix becomes

$$A_{k+1}^{-1} = \begin{pmatrix} \beta_k^{-1} & -\beta_k^{-1} v_k A_k^{-1} \\ -A_k^{-1} u_k \beta_k^{-1} & A_k^{-1} + A_k^{-1} u_k \beta_k^{-1} v_k A_k^{-1} \end{pmatrix}.$$

The solution z_{k+1} of the bordered system

$$A_{k+1} z_{k+1} = d_{k+1} = \begin{pmatrix} f_k \\ d_k \end{pmatrix}$$

can be computed by

$$z_{k+1} = \begin{pmatrix} 0 \\ z_k \end{pmatrix} + \begin{pmatrix} I_k \\ -A_k^{-1} u_k \end{pmatrix} \beta_k^{-1} \left(f_k - v_k z_k\right).$$

Similarly for the reverse bordering method, starting from

$$A_{k+1}^{-1} = \begin{array}{c} \begin{array}{cc} p_k' & n_k' \end{array} \\ \begin{pmatrix} a_k' & v_k' \\ u_k' & A_k' \end{pmatrix} \end{array} \begin{array}{c} p_k' \\ n_k' \end{array}$$

we have

$$A_k^{-1} = A_k' - u_k'\, a_k'^{-1}\, v_k'.$$

The solution z_k of the system

$$A_k z_k = d_k$$

can be obtained from the solution z_{k+1} of the bordered system

$$A_{k+1} z_{k+1} = d_{k+1} = \begin{array}{c} \begin{pmatrix} f_k \\ d_k \end{pmatrix} \end{array} \begin{array}{l} p'_k \\ n'_k \end{array}.$$

We set

$$z_{k+1} = \begin{pmatrix} c_k \\ z'_k \end{pmatrix} \begin{array}{l} p'_k \\ n'_k \end{array}$$

and we have

$$z_k = z'_k - u'_k \, a'^{-1}_k \, c_k.$$

Thus the block bordering method and the reverse bordering method can be applied in the following two cases.

Case 1.

$$\left(\begin{array}{c|c} A_k & u_k \\ \hline v_k & a_k \end{array}\right) z_{k+1} = \left(\begin{array}{c} d_k \\ \hline f_k \end{array}\right).$$

Case 2.

$$\left(\begin{array}{c|c} a_k & v_k \\ \hline u_k & A_k \end{array}\right) z_{k+1} = \left(\begin{array}{c} f_k \\ \hline d_k \end{array}\right).$$

There are also two other possibilities of bordering that could be investigated.

Case 3.

$$\left(\begin{array}{c|c} u_k & A_k \\ \hline a_k & v_k \end{array}\right) z_{k+1} = \left(\begin{array}{c} d_k \\ \hline f_k \end{array}\right).$$

This case can be treated the same as Case 2 because we can put the last p_k rows of the matrix and the right-hand side on the top and all the formulæ for the methods are the same.

Case 4.

$$\left(\begin{array}{c|c} v_k & a_k \\ \hline A_k & u_k \end{array}\right) z_{k+1} = \left(\begin{array}{c} f_k \\ \hline d_k \end{array}\right).$$

This case can be treated the same as Case 1 for the reason explained in Case 3.

Two particular cases can be interesting since they have many applications.

Let us first consider the case where A_k is a Hankel matrix; that is, when its elements a_{ij} are such that $a_{ij} = c_{i+j}$ where the c_i are given complex numbers. In this case, the reverse bordering method must be applied in its normal formulation because the structure of the inverse matrix does not permit any simplification.

Let us now consider the case of Toeplitz matrices. Let $A_k = (a_{ij})$ be the Hermitian positive definite Toeplitz matrix, built from a sequence of complex numbers $c_0, c_1, c_2, \ldots$. Thus we have $a_{ij} = c_{i-j}$ (for $i, j = 0, \ldots, n_k - 1$), $c_l = \bar{c}_{-l}$, and

$$A_k = T^{(0)}_{n_k} = \begin{pmatrix} c_0 & \bar{c}_1 & \bar{c}_2 & \cdots & \bar{c}_{n_k-1} \\ c_1 & c_0 & \bar{c}_1 & \cdots & \bar{c}_{n_k-2} \\ \vdots & \vdots & \vdots & & \vdots \\ c_{n_k-1} & c_{n_k-2} & c_{n_k-3} & \cdots & c_0 \end{pmatrix}.$$

In this case, A_{k+1} can be obtained from A_k by bordering either at the bottom and on the right or at the top and on the left. In both cases, due to the particular structure of the matrix, we have $v_k = \overline{u}_k^{\mathrm{T}}$.

Thus we can choose between two bordered matrices of dimension $n_{k+1} = n_k + p_k$

$$A_{k+1} = \begin{pmatrix} A_k & u_k \\ \overline{u}_k^{\mathrm{T}} & a_k \end{pmatrix} \quad \text{or} \quad A'_{k+1} = \begin{pmatrix} a_k & \overline{w}_k^{\mathrm{T}} \\ w_k & A_k \end{pmatrix}.$$

Obviously the two possibilities are not equivalent and the systems to be solved are different. However, in both cases, the bordering method can be applied after the simplification due to the special structure of the matrices.

If we consider the reverse unit matrix J (i.e., the unit matrix with its columns in the reverse order) of order n_k and the reverse unit matrix J' of order p_k, we have

$$w_k = J\overline{u}_k J'.$$

6. Numerical examples. When solving a system of linear equations by the bordering method some intermediate systems can be nearly singular. In that case, the block bordering method described in [5] allows us to jump over these near-singularities and the numerical stability of the process is thus improved.

Before giving a numerical example, let us discuss our strategy for deciding when and how far to jump. This strategy is based on the relation

$$\det A_{k+1} = \det A_k \cdot \det \beta_k.$$

Assuming that A_k^{-1} has already been obtained, we first add one new row and one new column to the matrix A_k; that is, we use the formulæ of §2 (or, equivalently, those of §3 with $p_k = 1$). If

$$|\beta_k| \leq \varepsilon$$

for some given $\varepsilon > 0$, we will add one more new row and one more new column to A_k and use the formulæ of §3 with $p_k = 2$. If

$$|\det \beta_k| \leq \varepsilon$$

we again add one new row and one new column to A_k and repeat the process until, after having added p_k new rows and columns, we obtain a matrix β_k such that

$$|\det \beta_k| > \varepsilon.$$

Then A_{k+1}^{-1} and z_{k+1} can be computed by the formulæ of §3. Let us also mention that the determinant of β_k is computed as the product of the pivots in a Gaussian elimination process. Such a strategy avoids the inversion of nearly singular matrices β_k, thus improving the numerical stability of the bordering method as shown by the following examples.

We first consider the system

$$\begin{pmatrix} 1 & 1 & 1 & 1 & -1 & 0 & -1 \\ 1 & 1 & 2 & 0 & 1 & 1 & -1 \\ 1 & 1 & -1 & 0 & 2 & -2 & 0 \\ -1 & 1 & 2 & 0 & -1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix} = \begin{pmatrix} -2 \\ 13 \\ -2 \\ 22 \\ 25 \\ 18 \\ 9 \end{pmatrix}.$$

TABLE 1
Solutions with the bordering method ($\eta = 10^{-14}$).

n							
1	−0.2000(1)						
2	−0.1501(16)	0.1501(16)					
3	−0.1001(16)	0.1001(16)	0.5000(1)				
4	−0.4500(1)	0.7500(1)	0.4992(1)	−0.1001(2)			
5	−0.2950(2)	−0.9297(1)	0.1333(2)	0.4833(2)	0.2500(2)		
6	−0.7006(15)	−0.1171(16)	0.9348(15)	0.1635(16)	0.7006(15)	−0.7006(15)	
7	0.9222	0.1859(1)	0.3006(1)	0.3570(1)	0.5078(1)	0.5922(1)	0.7000(1)

TABLE 2
Solutions with the block bordering method ($\eta = 10^{-14}, \varepsilon = 10^{-14}$).

n							
1	−0.2000(1)						
2	——	——					
3	——	——	——				
4	−0.4500(1)	0.7500(1)	0.5000(1)	−0.1000(2)			
5	−0.2950(2)	−0.9167(1)	0.1333(2)	0.4833(2)	0.2500(2)		
6	——	——	——	——	——	——	
7	0.1000(1)	0.2000(1)	0.3000(1)	0.4000(1)	0.5000(1)	0.6000(1)	0.7000(1)

In this system, the subsystems of dimensions 2, 3, and 6 are singular. Thus we add a perturbation η to a_{11} and a_{55}. So that the solution of the system remains the same, we also add η to the first component of the right-hand side and 5η to its fifth component.

Using the bordering and the block bordering methods for solving this system, we obtain the results in Tables 1 and 2 (ε denotes the threshold under which the block bordering method jumps and the numbers in parentheses denote the powers of 10).

However, in some applications, it is necessary to compute the solutions of all the intermediate systems. For example, this is the case in the computation of orthogonal polynomials [9], the Padé approximation, and the implementation of the progressive forms of extrapolation processes where the first step consists of the computation of the first descending diagonal of the triangular array [4]. In such cases, the block bordering method allows us to jump over the near-singular systems and then the reverse bordering method allows us to compute afterwards their solution with an improved numerical stability.

Let us first discuss the strategy used in the reverse bordering method. We assume that A_k^{-1} and A_{k+1}^{-1} are known and we want to compute the solutions of the intermediate systems of dimensions $n_{k+1}-1, \ldots, n_k+1$, which were skipped in the block bordering method when climbing to higher dimensions. We begin by deleting the last row and the last column of A_{k+1}^{-1}, that is, we use the formulæ of §4 with $p'_k = 1$ and $n'_k = n_{k+1} - 1$. For that, we must compute $a_k'^{-1}$. If a'_k is nearly singular, we delete the last two rows and the last two columns of A_{k+1}^{-1}; that is, we use the formulæ of §4 with $p'_k = 2$ and $n'_k = n_{k+1} - 2$ and so on until a nonnearly singular matrix a'_k has been obtained. The near singularity of a'_k is tested by computing its determinant (again by Gaussian elimination) and checking to see if $|\det a'_k| \leq \varepsilon'$ or not. However, if, in this test, we take $\varepsilon' \geq \varepsilon$ (where ε is the threshold used in the block bordering method) then a jump will occur from n_{k+1} to n_k and the intermediate systems that were not solved when climbing to higher dimensions will again be skipped. Thus we must choose $\varepsilon' < \varepsilon$.

TABLE 3

Solutions with the block bordering method and the reverse bordering method ($\eta = 10^{-14}, \varepsilon = 10^{-14}, \varepsilon' = 10^{-20}$).

n							
1	−0.2000(1)						
2	−0.1501(16)	0.1501(16)					
3	−0.1001(16)	0.1001(16)	0.5000(1)				
4	−0.4500(1)	0.7500(1)	0.5000(1)	−0.1000(2)			
5	−0.2950(2)	−0.9167(1)	0.1333(2)	0.4833(2)	0.2500(2)		
6	−0.7006(15)	−0.1168(16)	0.9341(15)	0.1635(16)	0.7006(15)	−0.7006(15)	
7	0.1000(1)	0.2000(1)	0.3000(1)	0.4000(1)	0.5000(1)	0.6000(1)	0.7000(1)

Let us now return to our previous example. Table 3 shows this improvement (ε' is the threshold for jumping in the reverse bordering method).

Let us now give an example with the ε-algorithm. This algorithm is an extrapolation process whose theory can be found in [4]. It can be interpreted as solving the system

$$\begin{cases} a_0 S_0 + a_1 S_1 + \cdots + a_k S_k &= 1 \\ a_0 S_1 + a_1 S_2 + \cdots + a_k S_{k+1} &= 1 \\ \vdots & \vdots \\ a_0 S_k + a_1 S_{k+1} + \cdots + a_k S_{2k} &= 1 \end{cases}$$

and then computing [1]

$$\varepsilon_{2k}^{(0)} = 1 / \sum_{i=0}^{k} a_i.$$

Let us apply the ε-algorithm to the partial sums of the series expansion of

$$f(x) = \frac{1 + b_1 x + \cdots + b_{m-1} x^{m-1} + x^m}{1 + x^m}.$$

Thanks to the theory of the ε-algorithm and its connection with Padé approximants (see the next section), we should have

$$\varepsilon_{2m}^{(0)} = f(x).$$

With $\eta = 0.25, m = 10, x = 2$, and $b_i = i \cdot \eta$, we have $f(x) = 2.998536585365854$. We set $\varepsilon = 10^{-6}$ and $\varepsilon' = 10^{-30}$ for the block bordering and reverse bordering methods and we obtain the following results for $\varepsilon_{2k}^{(0)}$. R means that the corresponding value was obtained by the reverse bordering method.

k	Bordering method	Block and reverse	
0	1.000000000000000	1.000000000000000	
1	0.833333333333333	0.833333333333333	
2	1.500000000000007	1.500000000000007	
3	1.500000000000006	1.500000000000007	R
4	1.500000000000004	1.499999999999987	R
5	1.500000000000004	1.499999999999984	R
6	1.500000000000006	1.500000000000006	R
7	1.500000000000007	1.500000000000006	
8	1.057370161706715	1.061205132114060	
9	5.442384375014805	5.530461077969034	
10	2.965829933964836	2.998536585365856	

Thus, only two exact digits are obtained for $\varepsilon_{10}^{(0)}$ with the bordering method and fifteen exact digits with the block bordering and reverse bordering methods.

Let us now take $\eta = 0.1, m = 10, x = 1, b_i = \eta/i$, and $\varepsilon = 10^{-3}$.

We have $f(x) = 1.141448412698413$ and we obtain

k	Bordering method	Block and reverse	
0	1.000000000000000	1.000000000000000	
1	1.200000000000000	1.200000000000000	
2	1.299999999999898	1.299999999999898	
3	1.366666666630687	1.366666666630683	R
4	1.416666663166007	1.416666665980086	R
5	1.416669182535733	1.416669185348150	R
6	1.370133070676631	1.370133070822927	R
7	1.363779438853716	1.363779438821982	
8	1.299241827763746	1.299241827748625	
9	1.221626694740750	1.221626694709201	
10	1.141448412733146	1.141448412698013	

Again the precision has been improved by the reverse bordering method.

In both examples, the subsystems of dimensions 3, 4, 5, and 6 were nearly singular and their solutions were obtained by the reverse bordering method from the solution of the system of dimension 7.

7. Application to Padé approximants. An important application of the bordering and the reverse bordering methods is the recursive computation of Padé approximants. We now recall the necessary definitions (see [2]).

A Padé approximant is a rational function whose series expansion in ascending powers of the variable agrees with a given power series f up to the term whose degree is the sum of the degrees of its numerator and its denominator. Such a Padé approximant is denoted by $[p/q]_f(x)$, where p is the degree of the numerator and q the degree of the denominator. By definition we have

$$[p/q]_f(x) - f(x) = O(x^{p+q+1}).$$

Let us define the linear functional c on the vector space of polynomials by

$$c\left(x^i\right) = c_i \quad \text{for } i = 0, 1, \ldots.$$

We consider the polynomial P_k of degree k belonging to the family of orthogonal polynomials with respect to c; that is,

$$P_k(x) = b_0 + b_1 x + \cdots + b_k x^k$$

such that

$$c\left(x^i\, P_k(x)\right) = \sum_{j=0}^{k} c_{i+j}\, b_j = 0 \quad \text{for } i = 0, 1, \ldots, k-1.$$

One of the b_i's is arbitrary and we choose the normalization $b_0 = 1$.

The coefficients $b_1, \ldots, b_k$ are obtained as the solution of a linear system

$$\begin{pmatrix} c_1 & c_2 & \cdots & c_k \\ c_2 & c_3 & \cdots & c_{k+1} \\ \vdots & \vdots & & \vdots \\ c_k & c_{k+1} & \cdots & c_{2k-1} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \end{pmatrix} = - \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \end{pmatrix}.$$

If we denote by A_k the Hankel matrix of the preceding system, by d_k its right-hand side, and by $z_k = (b_1, \ldots, b_k)^{\mathrm{T}}$ its solution, we can solve this system by the block bordering and the reverse bordering methods.

We consider the formal power series

$$f(x) = c_0 + c_1\, x + c_2\, x^2 + \cdots$$

and the polynomial $\widetilde{P}_k(x)$ given by

$$\widetilde{P}_k(x) = x^k\, P_k\left(x^{-1}\right) = b_k + b_{k-1}\, x + \cdots + b_1\, x^{k-1} + x^k.$$

From the connections between Padé approximants and orthogonal polynomials, we know that $\widetilde{P}_k(x)$ is the denominator of the Padé approximant $[k-1/k]_f\,(x)$. If we want to have the normalization $b_k = 1$ one can simply consider the polynomial

$$P_k^*(x) = b_k^{-1} \widetilde{P}_k(x).$$

Thus the block bordering and the reverse bordering methods allow us to compute recursively the coefficients of the denominators of the Padé approximants [0/1], [1/2], [2/3], ...

To control the accuracy and numerical stability of the reverse bordering method, we take f as the power series expansion of a rational function with numerator of degree $k-1$ and denominator of degree k. In that case, the Padé approximant $[k-1/k]_f$ must be identical to f. We set

$$f(x) = \frac{1 + \alpha_1 x + \alpha_2 x^2 + \cdots + \alpha_{k-1} x^{k-1}}{1 + \beta_0 x^k} = c_0 + c_1 x + c_2 x^2 + \cdots.$$

Giving some values to the α_i's and to the β_i's, we compute the c_i's so that $[k-1/k]_f = f$; that is, in order to have

$$P_k^* = 1 + \beta_0 x^k = 1 + \frac{b_0}{b_k} x^k,$$

where $b_0 = 1$, b_k is the coefficient of x^k in the orthogonal polynomial P_k, and all the $b_i, i = 1, \ldots, k-1$ are zero. The α_i's depend on a parameter η and we give to it different values.

We set ε as the threshold under which the block bordering method jumps, and ε' as the threshold under which the reverse bordering method jumps.

In the following examples, we give the residual $r_k = |A_k\, z_k - d_k|$, where z_k is the vector of the coefficients $b_i, i = 1, \ldots, k$ of the polynomial P_k (the numbers in parenthesis denote again the powers of 10). The coefficients of the polynomial P_k^*, which is the denominator of $[k-1/k] = f$, are also given.

7.1. Example 1. We consider the function

$$f(x) = \frac{1 + \eta x + \eta x^2}{1 - x^3} = 1 + \eta x + \eta x^2 + x^3 + \eta x^4 + \eta x^5 + x^6 + \cdots.$$

We should have $[2/3]_f = f$, that is, $P_3^* = 1 - x^3$.

$\eta = 10^{-5}$							
k	Bordering method no jump			Block and reverse $\varepsilon = 10^{-4}, \varepsilon' = 10^{-10}$			
1	.11(−15)			.0			R
2	.11(−15)	.46(−16)		.0	.16(−15)		R
3	.0	.21(−16)	.50(−11)	.22(−15)	.17(−20)	.0	

$P_3^* = 1 - .28 \cdot 10^{-16} x - .49 \cdot 10^{-11} x^2 - x^3$ $\quad P_3^* = 1 + .25 \cdot 10^{-21} x - .23 \cdot 10^{-20} x^2 - x^3$

$\eta = 10^{-10}$							
k	Bordering method no jump			Block and reverse $\varepsilon = 10^{-9}, \varepsilon' = 10^{-15}$			
1	.0			.0			R
2	.0	.83(−17)		.0	.83(−17)		R
3	.0	.25(−16)	.0	.0	.0	.0	

$P_3^* = 1 - .25 \cdot 10^{-16} x - x^3$ $\quad P_3^* = 1 + .61 \cdot 10^{-26} x - x^3$

7.2. Example 2. We consider the function

$$f(x) = \frac{1 + \eta x + 2\eta x^2 + 3\eta x^3}{1 - x^4} = 1 + \eta x + 2\eta x^2 + 3\eta x^3 + x^4 + \eta x^5 + 2\eta x^6 + 3\eta x^7 + \cdot$$

We should have $[3/4]_f = f$, that is, in particular, $P_4^* = 1 - x^4$.
The system to be solved is

$$\begin{pmatrix} \eta & 2\eta & 3\eta & 1 \\ 2\eta & 3\eta & 1 & \eta \\ 3\eta & 1 & \eta & 2\eta \\ 1 & \eta & 2\eta & 3\eta \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = - \begin{pmatrix} 1 \\ \eta \\ 2\eta \\ 3\eta \end{pmatrix}.$$

$\eta = 10^{-4}$				
k	Bordering method no jump			
1	.0			
2	.22(−15)	.51(−15)		
3	.0	.66(−15)	.47(−11)	
4	.44(−15)	.55(−15)	.25(−12)	.61(−12)

$P_4^* = 1 + .75 \cdot 10^{-15} x - .24 \cdot 10^{-12} x^2 - .61 \cdot 10^{-12} x^3 - x^4$

$\eta = 10^{-4}$					
k	Block and reverse $\varepsilon = 10^{-4}$, $\varepsilon' = 10^{-10}$				
1	.0				
2	.22(−15)	.51(−15)			R
3	.0	.23(−15)	.76(−16)		
4	.11(−15)	.20(−15)	.53(−15)	.61(−12)	

$P_4^* = 1 - .74 \cdot 10^{-16} x - .34 \cdot 10^{-15} x^2 - .61 \cdot 10^{-12} x^3 - x^4$

$\eta = 10^{-4}$					
k	Block and reverse $\varepsilon = 1.1 \cdot 10^{-4}$, $\varepsilon' = 10^{-10}$				
1	.67(−15)				R
2	.20(−12)	.17(−12)			R
3	.11(−15)	.23(−15)	.52(−15)		
4	.11(−15)	.12(−15)	.92(−16)	.61(−12)	

$P_4^* = 1 + .6 \cdot 10^{-17}x + .91 \cdot 10^{-16}x^2 - .61 \cdot 10^{-12}x^3 - x^4$

$\eta = 10^{-4}$					
k	Block and reverse $\varepsilon = 10^{-3}$, $\varepsilon' = 10^{-10}$				
1	.67(−15)				R
2	.67(−15)	.24(−15)			R
3	.11(−15)	.23(−15)	.76(−16)		R
4	.0	.0	.0	.0	

$P_4^* = 1 + .94 \cdot 10^{-21}x + .11 \cdot 10^{-19}x^2 - .24 \cdot 10^{-19}x^3 - x^4$

$\eta = 10^{-10}$				
k	Bordering method no jump			
1	.0			
2	.11(−15)	.26(−15)		
3	.0	.45(−15)	.35(−5)	
4	.22(−15)	.83(−16)	.36(−6)	.36(−16)

$P_4^* = 1 + .19 \cdot 10^{-15}x - .36 \cdot 10^{-6}x^2 - x^4$

$\eta = 10^{-10}$					
k	Block and reverse $\varepsilon = 10^{-9}$, $\varepsilon' = 10^{-15}$				
1	.0				R
2	.0	.37(−15)			R
3	.0	.83(−17)	.41(−15)		R
4	.0	.0	.0	.52(−25)	

$P_4^* = 1 - .37 \cdot 10^{-26}x - .93 \cdot 10^{-26}x^2 - .51 \cdot 10^{-25}x^3 - x^4$

$\eta = 10^{-15}$				
k	Bordering method no jump			
1	.0			
2	.0	.16(−15)		
3	.44(−15)	.12(−14)	.96(−2)	
4	.22(−15)	.11(−14)	.96(−2)	.96(−17)

$P_4^* = 1 + .1 \cdot 10^{-14}x + .96 \cdot 10^{-2}x^2 - x^4$

$\eta = 10^{-15}$					
k	Block and reverse $\varepsilon = 10^{-14}$, $\varepsilon' = 10^{-20}$				
1	.89(−15)				R
2	.11(−14)	.47(−15)			R
3	.22(−15)	.33(−15)	.32(−17)		R
4	.0	.20(−30)	.0	.0	

$P_4^* = 1 - .16 \cdot 10^{-30}x + .71 \cdot 10^{-31}x^2 - x^4$

7.3. Example 3. We consider the function

$$f(x) = \frac{1 + x + \eta x^2 + 2\eta x^3 + 3\eta x^4}{1 - x^5}$$
$$= 1 + x + \eta x^2 + 2\eta x^3 + 3\eta x^4 + x^5 + x^6 + \eta x^7 + 2\eta x^8 + 3\eta x^9 + \cdot$$

We should have $[4/5]_f = f$, that is, $P_5^* = 1 - x^5$.

$\eta = 10^{-5}$					
k	Bordering method no jump				
1	.0				
2	.0	.0			
3	.22(−15)	.0	.29(−15)		
4	.22(−15)	.22(−15)	.51(−15)	.62(−11)	
5	.22(−15)	.22(−15)	.47(−15)	.84(−11)	.11(−15)

$P_5^* = 1 - .41 \cdot 10^{-11}x + .41 \cdot 10^{-11}x^2 + .41 \cdot 10^{-11}x^3 - .41 \cdot 10^{-11}x^4 - x^5$

$\eta = 10^{-5}$						
k	Block and reverse $\varepsilon = 10^{-4}$, $\varepsilon' = 10^{-10}$					
1	.0					
2	.0	.0				R
3	.11(−15)	.0	.16(−15)			R
4	.0	.0	.46(−16)	.73(−16)		
5	.22(−15)	.0	.61(−16)	.26(−15)	.30(−15)	

$P_5^* = 1 - .15 \cdot 10^{-16}x + .76 \cdot 10^{-16}x^2 + .18 \cdot 10^{-15}x^3 + .12 \cdot 10^{-15}x^4 - x^5$

$\eta = 10^{-15}$					
k	Bordering method no jump				
1	.0				
2	.0	.0			
3	.0	.22(−15)	.11(−15)		
4	.22(−15)	.11(−15)	.11(−15)	.55(−1)	
5	.22(−15)	.11(−15)	.15(−17)	.11	.59(−15)

$P_5^* = 1 + .6 \cdot 10^{-1}x - .6 \cdot 10^{-1}x^2 - .6 \cdot 10^{-1}x^3 + .6 \cdot 10^{-1}x^4 - 1.06x^5$

$\eta = 10^{-15}$						
k	Block and reverse $\varepsilon = 10^{-14}$, $\varepsilon' = 10^{-20}$					
1	.0					
2	.0	.22(−15)				R
3	.11(−15)	.22(−15)	.11(−15)			R
4	.0	.11(−15)	.33(−15)	.33(−15)		
5	.0	.0	.11(−15)	.44(−15)	.22(−15)	

$$P_5^* = 1 + .11 \cdot 10^{-15}x - .22 \cdot 10^{-15}x^2 - .22 \cdot 10^{-15}x^3 - x^5$$

8. Conclusions. In Gaussian elimination, pivotal strategies are often necessary to ensure a better numerical stability. In particular, they avoid division by numbers close to zero (which are possibly due to cancellation errors in the previous steps), thus preventing possible catastrophic errors. The block bordering method provides a similar strategy in a different context for the same drawback. However, with such a strategy, the solutions of the intermediate systems that were skipped when climbing to higher and higher dimensions are not computed. It was the purpose of this paper to propose an algorithm (the reverse bordering method) for obtaining these solutions.

REFERENCES

[1] C. Brezinski, *Résultats sur les procédés de sommation et sur l'ε-algorithme*, RIRO, R3 (1970), pp. 147–153.
[2] ———, *Padé-type Approximation and General Orthogonal Polynomials*, Birkhäuser-Verlag, Basel, 1980.
[3] ———, *Bordering methods and progressive forms for sequence transformations*, Zastos. Mat., 20 (1990), pp. 435–443.
[4] C. Brezinski and M. Redivo-Zaglia, *Extrapolation Methods. Theory and Practice*, North-Holland, Amsterdam, 1991.
[5] C. Brezinski, M. Redivo-Zaglia, and H. Sadok, *A breakdown-free Lanczos type algorithm for solving linear systems*, Numer. Math., 63 (1992), pp. 29–38.
[6] W. J. Duncan, *Some devices for the solution of large sets of simultaneous linear equations*, Philos. Mag. Ser. 7, 35 (1944), pp. 660–670.
[7] V. N. Faddeeva, *Computational Methods of Linear Algebra*, Dover, New York, 1959.
[8] W. W. Hager, *Updating the inverse of a matrix*, SIAM Rev., 31 (1989), pp. 221–239.
[9] M. M. Cecchi and M. Redivo-Zaglia, *A new recursive algorithm for a Gaussian quadrature formula via orthogonal polynomials*, in Orthogonal Polynomials and Their Applications, C. Brezinski, L. Gori, and A. Ronveaux, eds., Baltzer, Basel, 1991, pp. 353–358.