

On the solution of general Bordered ABD linear systems

Pierluigi Amodio^{*1}, Ian Gladwell², and Giuseppe Romanazzi¹

¹ Dipartimento di Matematica, Università di Bari, I-70125 Bari, Italy

² Department of Mathematics, Southern Methodist University, 75275-0156 Dallas, Texas, USA

Key words boundary value problems, linear systems, bordered ABD matrices.

2000 Mathematics Subject Classification 65F05, 65F50, 65L10

In this paper we analyze the solution of Bordered Almost Block Diagonal (BABD) linear systems arising from the discretization of BVPs with nonseparated boundary conditions using some well known numerical methods, for example spline collocation at Gaussian points. Several approaches based on cyclic reduction are proposed and compared.

© 2005 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim

1 Introduction

One of the most expensive parts of any package for the solution of boundary value problems (BVPs) is the solution of the linear systems generated internally. So, much of the effort in constructing such codes, is first devoted to defining numerical schemes in order to obtain linear systems with well defined and simple sparsity structure, and, second to minimizing the number of generated meshes, which is proportional to the number of matrix factorizations.

Each package for BVPs uses well established codes for the solution of sparse structured linear systems. Among the others, SOLVEBLOK [5] is widely used since it solves quite general block banded linear systems. Since some BVP codes require the solution of so-called Almost Block Diagonal (ABD) systems, specific codes constructed for such systems, such as COLROW [6], ARCECO[6] and ABDPACK [7] are often preferred. Because of their linear algebra requirements, these codes can only be applied to BVPs with separated boundary conditions. The discretization of BVPs with non-separated boundary conditions leads to Bordered Almost Block Diagonal (BABD) matrices, with one additional block which is completely outside the ABD structure (either in the left-lower corner or in the right-upper). Though it is possible to rearrange BVPs with non-separated BCs into BVPs of double the size with separated BCs, in [3] it is shown that this is not always efficient.

For this reason in [2], based on an idea in [1], a Fortran 90 code, BABDCR, was proposed for the solution of BABD matrices. This code solves BABD systems which can be seen as block bidiagonal matrices with an additional corner block. In [3], BABDCR has been successfully applied in the codes MIRKDC and PMIRKDC. Here, we generalize the cyclic reduction approach in [1, 2] to the solution of general BABD linear systems

$$Ax = f \tag{1}$$

where the structure of the coefficient matrix A is depicted in Figure 1.

ABD matrices are characterized by a very special sparsity pattern: the nonzero elements are grouped in block rows; there is no intersection between the nonzero columns of two non consecutive block rows; finally, the main diagonal entries always lie in the nonzero blocks. BABD matrices satisfy an additional property: the last (or the first) block row has an extra block in the left-lower (right-upper) corner whose nonzero columns only intersect the nonzero columns of the first (last) block row. The blocks in this row are called *boundary blocks* since they arise due to the boundary conditions when solving a BVP.

Since each block row of the BABD matrix has columns overlapping with the previous and the successive block rows, and there is no intersection between these overlapped columns, we recognize three blocks in each block

* Corresponding author: e-mail: amodio@dm.uniba.it, Phone: +39 080 5442703, Fax: +39 080 5460612.

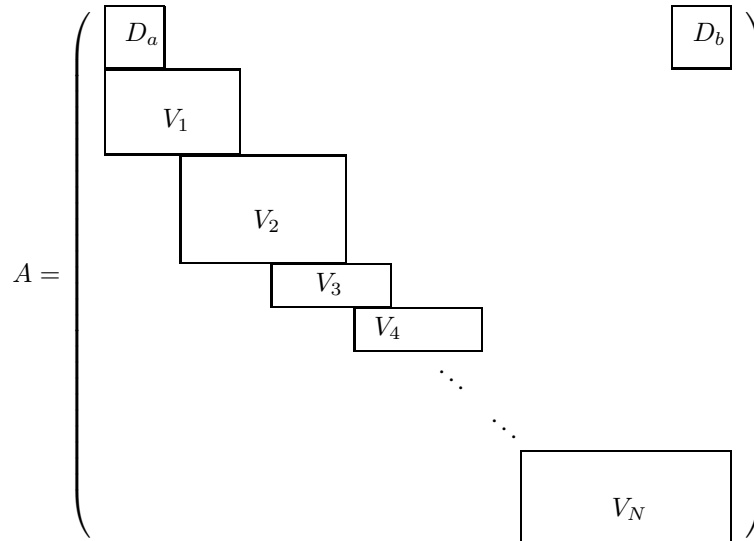


Fig. 1 Structure of a general BABD matrix.

row

$$V_i = \begin{pmatrix} A_i & B_i & C_i \end{pmatrix}$$

where the columns of A_i and C_i overlap the columns of V_{i-1} and V_{i+1} , respectively. Moreover, V_1 and V_N have the same structure as each other V_i : A_1 is overlapped with D_a and C_N is overlapped with D_b . Matrices with this shape arise, for example, from the numerical solution of two-point boundary value problems with non-separated boundary conditions when using spline collocation with B-splines or monomial splines at Gaussian points (see [4]). The idea is to determine a piecewise polynomial approximation to the exact solution satisfying continuity conditions on a given mesh. The blocks V_i have size $(n+k) \times (2n+k)$ and boundary blocks have size $n \times n$, where n represents the order of the ODE and k the number of Gaussian points in each subinterval of the mesh. In addition, when monomial splines are used,

$$A_i = \begin{pmatrix} H_i \\ -C_i \end{pmatrix}, \quad B_i = \begin{pmatrix} G_i \\ -D_i \end{pmatrix} \quad \text{and} \quad C_i = \begin{pmatrix} O \\ I \end{pmatrix},$$

where I is an identity matrix of size n , and O is a $k \times n$ null matrix.

In general, suppose that V_i has m_i rows while A_i and B_i have n_i and k_i columns, respectively. Then each block V_i is $m_i \times (n_i + k_i + n_{i+1})$ and is overlapped by V_{i-1} in the first n_i columns and by V_{i+1} in the last n_{i+1} columns. Blocks D_a and D_b have dimensions $m_{BC} \times n_1$ and $m_{BC} \times n_{N+1}$, respectively. Therefore, we must have

$$m_{BC} + \sum_{i=1}^N m_i = \sum_{i=1}^{N+1} n_i + \sum_{i=1}^N k_i.$$

To satisfy the properties of a structurally nonsingular BABD matrix, the following conditions must also be satisfied:

$$m_i + m_{i+1} \geq k_i + n_{i+1} + k_{i+1}, \quad i = 1, \dots, N-1 \quad (2)$$

$$n_1 + n_{N+1} \geq m_{BC}.$$

Since in block form the coefficient matrix has $N+1$ rows and $2N+1$ columns, in the next section we use for simplicity the following definitions for the right hand side and for the unknown vector:

$$f = (d^T, f_1^T, \dots, f_N^T)^T, \quad x = (z_1^T, w_1^T, z_2^T, w_2^T, \dots, z_N^T, w_N^T, z_{N+1}^T)^T.$$

2 The cyclic reduction algorithm

The idea (see [1]) is to cyclically reduce (1) in order to derive systems of lower dimension (involving less unknowns) with the same BABD structure. At the first step of reduction, for $i = 1, 3, 5, \dots$, two consecutive block rows

$$\begin{aligned} A_i z_i + B_i w_i + C_i z_{i+1} &= f_i \\ A_{i+1} z_{i+1} + B_{i+1} w_{i+1} + C_{i+1} z_{i+2} &= f_{i+1} \end{aligned} \quad (3)$$

are coupled to obtain the row

$$A'_i z_i + B'_i w'_i + C'_{i+1} z_{i+2} = f'_i \quad (4)$$

involving the unknowns z_i and z_{i+2} , and a new unknown w'_i defined below.

Blocks A'_i and C'_{i+1} in (4) have the same number of columns as A_i and C_{i+1} , respectively. Moreover, A'_i overlaps the columns of C'_{i-1} obtained by the reduction of the previous pair of rows (A'_1 overlaps D_a), and C'_{i+1} overlaps the columns of the matrix A'_{i+2} obtained by the reduction of the next pair of block rows (C'_N overlaps D_b). If N is odd, the last row is unchanged in the reduced matrix. Therefore, the number of block rows of the new matrix is $\lceil N/2 \rceil + 1$.

The size of equation (4) strongly depends on the choice of matrix M_i used to obtain (4) from (3). M_i is chosen as one of the matrices:

1. $M_i = \begin{pmatrix} B_i & C_i \\ & A_{i+1} & B_{i+1} \end{pmatrix}$ leading to a null vector w'_i (since $B'_i = O$);
2. $M_i = \begin{pmatrix} C_i \\ A_{i+1} & B_{i+1} \end{pmatrix}$ leading to $w'_i = w_i$;
3. $M_i = \begin{pmatrix} B_i & C_i \\ & A_{i+1} \end{pmatrix}$ leading to $w'_i = w_{i+1}$;
4. $M_i = \begin{pmatrix} C_i \\ A_{i+1} \end{pmatrix}$ leading to $w'_i = \begin{pmatrix} w_i \\ w_{i+1} \end{pmatrix}$.

The choice among the various M_i depends strongly on the sparsity structure of the blocks involved. The choice of M_i in 4. rather than in 1. reduces the number of operations to compute the factorization of M_i but gives a reduced system of larger size. If $k_i = 0$ for any i , then the only available choice 4. has been investigated in detail in [1, 2]. Because of (2) and the non-singularity of A , we can set M_i to size equal to $(m_i + m_{i+1}) \times s_i$ and rank s_i . Then, the number of equations in (4) is $m'_i = m_i + m_{i+1} - s_i$ and B'_i is $m'_i \times (k_i + n_{i+1} + k_{i+1} - s_i)$. That is, the reduced system does not involve at least the unknowns z_i , $i = 2, 4, 6, \dots$, and its size is reduced by

$$\sum_{i=1}^{\lfloor N/2 \rfloor} s_{2i-1}. \text{ In general from}$$

$$P_i M_i = \begin{pmatrix} L_i \\ S_i \end{pmatrix} U_i \equiv \begin{pmatrix} L_i & \\ S_i & I \end{pmatrix} \begin{pmatrix} U_i \\ O \end{pmatrix},$$

where P_i is a $(m_i + m_{i+1}) \times (m_i + m_{i+1})$ permutation matrix and L_i, U_i are the triangular matrices of the corresponding LU factorization, the reduced system is obtained by multiplying (3) by P_i and the inverse of the lower triangular matrix of the M_i factorization. For example, in case of M_i in 4., one has

$$\begin{pmatrix} L_i \\ S_i & I \end{pmatrix}^{-1} P_i \begin{pmatrix} A_i & B_i & C_i \\ & A_{i+1} & B_{i+1} & C_{i+1} \end{pmatrix} = \begin{pmatrix} \tilde{A}_i & \tilde{B}_i & U_i & \tilde{B}_{i+1} & \tilde{C}_{i+1} \\ A'_i & B'_i & & B'_{i+1} & C'_{i+1} \end{pmatrix} \quad (5)$$

where B'_i in (4) equals $\begin{pmatrix} B'_i & B'_{i+1} \end{pmatrix}$ in (5). The right hand side is modified accordingly. For the other choices of M_i the procedure is obviously simplified since B_i and/or B_{i+1} are included in M_i and thus do not appear in the reduced system.

For the factorization of M_i , suppose for simplicity $k_i = k \neq 0$, $n_i = n$ and $m_i = n + k$, and consider the factorization of M_i in 1. (for the other cases the factorization may be easily deduced). There are two stages: in the first, k steps of factorization are performed independently on $(n + k) \times k$ blocks B_i and B_{i+1} thus obtaining

$$\begin{pmatrix} Q_i & \\ & Q_{i+1} \end{pmatrix} \begin{pmatrix} B_i & C_i \\ & A_{i+1} & B_{i+1} \end{pmatrix} = \begin{pmatrix} \tilde{L}_i & & \\ S_i & I & \\ & & \tilde{L}_{i+1} \\ & & S_{i+1} & I \end{pmatrix} \begin{pmatrix} \tilde{U}_i & T_i \\ & \tilde{C}_i \\ T_{i+1} & \tilde{U}_{i+1} \\ \tilde{A}_{i+1} & \end{pmatrix},$$

where \tilde{L} and \tilde{U} are $k \times k$ triangular matrices, I is the identity matrix of size n , Q are permutation matrices, and the other blocks are defined accordingly. In the second stage, n steps of factorization are performed on $\begin{pmatrix} \tilde{C}_i \\ \tilde{A}_{i+1} \end{pmatrix}$ thus obtaining

$$\bar{Q} \begin{pmatrix} \tilde{U}_i & T_i \\ & \tilde{C}_i \\ T_{i+1} & \tilde{U}_{i+1} \\ \tilde{A}_{i+1} & \end{pmatrix} = \begin{pmatrix} I & & \\ & \bar{L}_i & \\ & & I \\ \bar{S}_i & & & I \end{pmatrix} \begin{pmatrix} \tilde{U}_i & T_i \\ & \tilde{U}_i \\ T_{i+1} & \tilde{U}_{i+1} \\ \tilde{A}_{i+1} & \end{pmatrix}.$$

The lower triangular matrix must be non singular. This is the reason of the fourth null row of the upper triangular matrix (and the identity matrix in the fourth column of the lower triangular matrix). Then, the reduced system in (5) is obtained by pre-multiplying

$$\begin{pmatrix} I & & \\ & \bar{L}_i & \\ & & I \\ \bar{S}_i & & & I \end{pmatrix}^{-1} \bar{Q} \begin{pmatrix} \tilde{L}_i & & \\ S_i & I & \\ & & \tilde{L}_{i+1} \\ & & S_{i+1} & I \end{pmatrix}^{-1} \begin{pmatrix} Q_i & \\ & Q_{i+1} \end{pmatrix}$$

to the equations in (3). None of the factorizations require fill-in. Additional storage is however required for the successive reduced matrices if (as is usual in the algorithms devoted to BVP solution) the factorization is computed once and used to solve several linear systems.

The reduction must be iterated until, after $p = \log_2(N)$ steps, the 2×2 block linear system

$$\begin{pmatrix} D_a & D_b \\ A_1^{(p)} & C_N^{(p)} \end{pmatrix} \begin{pmatrix} z_1 \\ z_{N+1} \end{pmatrix} = \begin{pmatrix} d \\ f_1^{(p)} \end{pmatrix} \quad (6)$$

(of size $m_1 + m_{N+1}$) is obtained. Then, the algorithm for the solution of (1) proceeds with a back-substitution phase that first computes z_1 and z_{N+1} from (6), and then the other unknowns in reverse order.

References

- [1] P. Amodio and M. Paprzycki, *SIAM J. Numer. Anal.* **18**, 56–68 (1997).
- [2] P. Amodio and G. Romanazzi, Report of the Dipartimento di Matematica, Università di Bari **40/04** (2004).
- [3] P. Amodio et al., Preprint Dipartimento di Matematica, Università di Bari (2005).
- [4] U. Ascher et al., *SIAM J. Numer. Anal.* **20** (no. 1), 121–142 (1983)
- [5] C. De Boor and R. Weiss, *ACM Trans. Math. Softw.* **6** (no. 1), 80–87 (1980); *ACM Trans. Math. Softw.* **6** (no. 1), 88–91 (1980).
- [6] J. C. Diaz et al., *ACM Trans. Math. Softw.* **9** (no. 3), 358–375 (1983); *ACM Trans. Math. Softw.* **9** (no. 3), 376–380 (1983).
- [7] F. Majaess et al., *ACM Trans. Math. Softw.* **18** (no. 2), 193–204 (1992); *ACM Trans. Math. Softw.* **18** (no. 2), 205–210 (1992).