



Biconjugate Decomposition Using ABS Algorithms

E. Golpar-Raboky ^{a,*}, F. Fattahzadeh ^b, T. Shojaeizadeh ^a

(a) *Qom University , Qom, Iran*

(b) *Islamic Azad University (Central Tehran Branch)*

(b) *Isfahan University of Technology , Isfahan, Iran*

Abstract

ABS method provides the general solution of a system by computing a solution and a matrix, called the Abaffian matrix, with rows generating the null space of the coefficient matrix. We present an algorithm for computing a biconjugate pair (V,P) , such that $V^TAP = \Omega$ is a diagonal and nonsingular matrix, using ABS algorithm. Then we propose an algorithm for computing an equivalent diagonal form of a matrix A by using the extended ABS algorithm.

Keywords : ABS algorithms; Extended ABS algorithm; Biconjugate pair; Matrix decomposition; Biconjugate decomposition; Equivalent diagonal form.

1 Introduction

ABS methods constitute a large class of methods, first introduced by Abaffy et al. [1], for solving linear algebraic systems, and later extended to solve least square problems, nonlinear algebraic equations, optimization problems [2, 8, 9] and recently to Diophantine systems [5, 7]. ABS methods are a direct iterative class of methods for solving linear equations. Each method in the class provides the general solution of the system by computing a particular solution and a matrix, the Abaffian matrix, with rows generating the null space of the coefficient matrix. The method starts with an initial vector $x_1 \in R^n$ (arbitrary) and a nonsingular matrix $H_1 \in R^{n \times n}$ (Spedicato's parameter). Given x_i as a solution of the first $i - 1$ equations, and the Abaffian matrix H_i with rows generating the null space of the first $i - 1$ equations, the ABS algorithm computes x_{i+1} and H_{i+1} as the solution and null space generator of the first i equations, respectively. The choices of

*Corresponding author. Email address: g_raboky@mehr.sharif.edu

- (1) Choose $H_1 \in R^{n \times n}$, arbitrary and nonsingular. Let $i=1$, and $r_i = 0$.
- (2) Compute $s_i = H_i a_i$.
- (3) **If** $s_i = 0$ **and** , **then** let $H_{i+1} = H_i$, $r_{i+1} = r_i$ and **go to** step (6).
- (4) Compute search vector p_i by $p_i = H_i^T z_i$, where $z_i \in R^n$ is arbitrary save for the condition $a_i^T H_i^T z_i \neq 0$.
- (5) Update matrix H_i by,
$$H_{i+1} = H_i - \frac{H_i a_i w_i^T H_i}{w_i^T H_i a_i}$$
where $w_i \in R^n$ is arbitrary save for the condition $a_i^T H_i^T w_i \neq 0$.
- (6) **If** $i = m$ **then Stop** (r_{m+1} is the rank of A), **else** let $i = i + 1$ and **go to** step (2).

Matrices H_i , which are generalizations of projection matrices, have been called Abaffians.

Chen et al. [4] introduced a generalization of the ABS algorithms, called extended ABS (EABS) class of algorithms for the real case, which differs from the ABS algorithms only in updating the Abaffian matrices H_i . In the EABS algorithms, the Abaffian matrices H_i are updated as follows:

- $H_{i+1} = G_i H_i$, where $G_i \in R^{j_{i+1} \times j_i}$ is such that we have $G_i x = 0$ if and only if $x = \lambda H_i a_i$, for some $\lambda \in R$.

For the (not necessarily independent) rows of H_{i+1} to be the generator of the null space of the first i rows of A , we must have $\text{rank}(G_i) \geq n - r_{i+1}$, where r_{i+1} is the rank of the first i rows of A . It can be easily verified that if in the EABS algorithm, we let $j_1 = \dots = j_i = \dots = n$ and $G_i = I - H_i a_i w_i^T / w_i^T H_i a_i$, $1 \leq i \leq m$, where w_i satisfies $w_i^T H_i a_i \neq 0$, then the EABS algorithm turns into a basic ABS algorithm. Furthermore, as in the ABS algorithms, in the EABS algorithms, for every i , $1 \leq i \leq m$, we have $H_i a_i \neq 0$ if and only if a_i is linearly independent of a_1, a_2, \dots, a_{i-1} . Indeed, the general solution of the first $i - 1$ equations of the system is $x_i + H_i^T y_i$, $y_i \in R^{j_i}$ (see [4]).

Remark 2.1. *We can see that the extended ABS algorithms can always be tuned to produce a basis for the null space of the coefficient matrix. Let $G_i \in R^{n-r_i \times n-r_i+1}$, then, $H_{i+1} = G_i H_i$ is a full row rank matrix and generates a basis for the null space of the first i rows of A .*

We recall some properties of the ABS class, assuming that A has full rank.

p1: The vector $H_i a_i$ is zero if and only if a_i is linearly dependent on a_1, \dots, a_{i-1} .

p2: The vector $H_i^T w_i$ is zero if and only if w_i is linearly dependent on w_1, \dots, w_{i-1} .

p3: Define $A_i = (a_1, \dots, a_i)$ and $W_i = (w_1, \dots, w_i)$. Then,

$$H_{i+1} A_i = 0, \quad H_{i+1}^T W_i = 0. \quad (2.2)$$

$$MAP = L = \begin{pmatrix} l \\ 0 \end{pmatrix}, \quad (3.6)$$

where matrix $l \in R^{r \times r}$ is a nonsingular lower triangular matrix.

Here, we state a theorem and then show how to choose parameters of the ABS algorithm, in **phase 2** to compute a biconjugate decomposition of A .

Theorem 3.1. *Let $A \in R^{n \times n}$ be strongly nonsingular (i.e., all principal submatrices are nonsingular). Then, the choices $H_1 = I$ and $w_i = e_i$ are well defined and the following properties hold:*

- (a) *The first r rows of H_{i+1} are identically zero.*
- (b) *The last $n - i$ columns of H_{i+1} are equal to the last $n - i$ columns of H_1 .*
- (c) *$P = (p_1, \dots, p_n)$ is an upper triangular matrix.*

Proof. See [2].

Let $B = L^T$, since the submatrix l is strongly nonsingular. Now, we apply the ABS algorithm with coefficient matrix B . By Theorem (3.1), we can compute Abaffian matrices and search vectors as follow:

Let $R_1 = I_{m,m}$, update R_i by,

$$R_{i+1} = R_i - \frac{R_i b_i e_i^T H_i}{e_i^T R_i b_i},$$

where b_i is the i th row of B , for $i = 1, \dots, r$. Let $Q = (q_1, \dots, q_r)$ where $q_i = R_i^T e_i$. According to Theorem (3.1), Q is an upper triangular matrix thus, BQ is also an upper triangular matrix and by ABS properties (**p4**), BQ is a lower triangular matrix, therefore, BQ is a nonsingular and diagonal matrix and, we have,

$$\Omega = Q^T B^T = Q^T MAP = V^T AP,$$

is a biconjugate decomposition of A , where, $V = M^T Q$. Moreover, we have,

$$\text{rank}(A) = \text{rank}(V) = \text{rank}(P) = r, \quad V \in R^{m \times r}, \quad P \in R^{n \times r}.$$

Now, we ready to present an algorithm.

Algorithm 2. A biconjugate decomposition by ABS algorithm

First Phase:

- (1) Choose $H_1 \in R^{n \times n}$, arbitrary and nonsingular, and $M = I_{m,m}$. Let $i=1$, and $r_i = 0$.
- (2) Compute $s_i = H_i a_i$. **If** $s_i = 0$, **then** let $H_{i+1} = H_i$, $r_{i+1} = r_i$, shift the i th row of A and M to the end of the matrices, and **go to** step (5) (the i th equation is redundant) **else** $r_{i+1} = r_i + 1$.

(3) Compute $b_i = Ap_i$ and search vector v_i by, $v_i = R_i^T e_i$, where $e_i \in R^m$ is the i th unit vector.

(4) Update matrices H_i and R_i by

$$H_{i+1} = H_i - \frac{H_i a_i w_i^T H_i}{w_i^T H_i a_i}$$

where $w_i \in R^n$ is arbitrary save for the condition $a_i^T H_i^T w_i \neq 0$, and

$$R_{i+1} = R_i - \frac{R_i b_i e_i^T R_i}{e_i^T R_i b_i}.$$

(5) If $i < m$, then let $i = i + 1$, go to step (2), else let

$V = (v_1, \dots, v_m)$ and $P = (p_1, \dots, p_m)$. Then

$$\Omega = V^T AP,$$

is a biconjugate decomposition of A .

(6) Stop.

An equivalent diagonal of a matrix by ABS algorithm

Definition 3.2. Matrices $A, D \in R^{m,n}$ are said to be equivalent if there exist nonsingular matrices $V \in R^{m,m}$ and $U \in R^{n,n}$, such that

$$V^T AU = D \tag{3.7}$$

Here, we present a two-phase algorithm based on the extended ABS algorithm for computing nonsingular matrices V and U , such that $V^T AU \in R^{m \times n}$ is a diagonal matrix. Assume that $A \in R^{m \times n}$ is of rank r . For the aim of computation of nonsingular matrices V and U , in the first phase we apply extended the ABS algorithm and obtain the full rank matrix H_{m+1} as a basis for the null space of MA (M is a permutation matrix such that the first r rows of MA are independent). Then, $U = (p_1, \dots, p_r, H_{m+1}^T)$ is a nonsingular matrix and $MAU = L \in R^{m \times n}$ is a lower triangular matrix. Let $B = L^T$. Now, apply the ABS algorithm with coefficient matrix B . By Theorem (3.1), we can compute Abaffian matrices and search vectors as follows:

Let $R_1 = I_{m,m}$, update R_i by,

$$R_{i+1} = R_i - \frac{R_i b_i e_i^T H_i}{e_i^T R_i b_i},$$

where b_i is the i th row of B , for $i = 1, \dots, r$. According to Theorem (3.1), first r rows of R_{m+1} equal zero, then we delete the zero rows for generating a basis for the null space of B . Let $Q = (q_1, \dots, q_r, R_{r+1}^T)$, where $q_i = R_i^T e_i$, then Q is a nonsingular upper triangular matrix and BQ is a diagonal matrix. Therefore,

$$D = Q^T B^T = Q^T MAU = V^T AU,$$

4 Examples

In this section, we compute a biconjugate decomposition of matrix A using the proposed algorithms.

Example 4.1. Consider the following matrix:

$$A = \begin{bmatrix} 75 & 50 & 75 & 100 & 50 \\ 50 & 50 & 100 & 75 & 100 \\ 100 & 50 & 50 & 50 & 50 \\ 25 & 75 & 50 & 100 & 25 \\ 75 & 25 & 100 & 100 & 50 \end{bmatrix}.$$

Upon an application of our proposed algorithm to compute a biconjugate decomposition of A , we obtain the following results.

$$V = \begin{pmatrix} 1 & -0.7 & -2 & 1 & -1.7 \\ 0 & 1 & 1 & -3 & -0.3 \\ 0 & 0 & 1 & 0.5 & 0 \\ 0 & 0 & 0 & 1 & 0.7 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

and

$$U = \begin{pmatrix} 1 & -0.7 & -1 & 1 & 0 \\ 0 & 1 & -0.5 & -3 & -0.3 \\ 0 & 0 & 0 & 1 & -1.3 \\ 0 & 0 & 1 & 0 & 0.7 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Then

$$\Omega = V^T A U = \begin{pmatrix} 75 & 0 & 0 & 0 & 0 \\ 0 & 16.7 & 0 & 0 & 0 \\ 0 & 0 & -75 & 0 & 0 \\ 0 & 0 & 0 & -150 & 0 \\ 0 & 0 & 0 & 0 & -25 \end{pmatrix},$$

is an biconjugate decomposition of A .

5 Conclusion

In this paper, we presented some algorithms for computing an diagonal form of matrix A based on ABS algorithms. We presented a two-phase algorithm for computing a biconjugate decomposition of a matrix A with arbitrary rank, making use of ABS algorithms. Also, we proposed an algorithm for computing an equivalent diagonal form of a matrix A using the extended ABS algorithm.

References

- [1] J. Abaffy, C.G Broyden, E. Spedicato, A class of direct methods for linear systems, Numerische Mathematik 45 (1984) 361-376.

