Extended reduced rank two Abaffian update schemes in the ABS-type methods

K. Amini b, N. Mahdavi-Amiri a,*, M.R. Peyghami a

a Department of Mathematical Sciences, Sharif University of Technology, P.O. Box 11365-9415, Tehran, Iran
b Department of Sciences, Razi University, Kermanshah, Iran

Abstract

The ABS methods, introduced by Abaffy, Broyden and Spedicato, are direct iteration methods for solving a linear system where the \(i\)th iterate satisfies the first \(i\) equations, therefore a system of \(m\) equations is solved in at most \(m\) steps. Recently, we have presented a new approach to devise a class of ABS-type methods for solving full row rank systems [K. Amini, N. Mahdavi-Amiri, M. R. Peyghami, ABS-type methods for solving full row rank linear systems using a new rank two update, Bulletin of the Australian Mathematical Society 69 (2004) 17–31], the \(i\)th iterate of which solves the first \(2i\) equations. Here, to reduce the space and computation time, we use a new extended rank two update formula for the Abaffian matrix so that the number of rows of the Abaffian matrix is reduced by two in every iteration. This extension along with the reduction offer more flexibility for the definition of the Abaffian matrix.

© 2006 Elsevier Inc. All rights reserved.

Keywords: ABS methods; Rank two update; Linear system; Abaffian update

1. Introduction

The ABS methods, introduced by Abaffy, Broyden and Spedicato [1,2], constitute a general class of algorithms for solving linear and nonlinear algebraic systems. The basic algorithm works on a system of the form

\[Ax = b,\]  

(1.1)

where \(A \in \mathbb{R}^{m \times n}\), \(x \in \mathbb{R}^n\) and \(b \in \mathbb{R}^m\). The basic ABS methods determine the solution of (1.1) or signify its lack of existence in at most \(m\) iterations. In any iteration, one extra equation, if compatible, is satisfied.

A major step in an ABS algorithm is concerned with the definition of the so-called Abaffian matrix. To reduce the space and computation time of the algorithms, various ways have been suggested to reduce the size of the Abaffian in every iteration [3].
We have recently devised a new approach, based on ABS methods, which satisfies two new equations, if compatible, in every iteration and makes use of a new rank two update formula for the Abaffian matrix [3]. Simple schemes to reduce rows of the Abaffian matrix after a rank two update can be seen in [3].

Here, in order to offer more flexibility in the algorithms and to save the time and space as well, we give an extended definition of the Abaffian matrix in our approach. By an appropriate choice of the parameters in our new extended rank two update formula, two rows of the Abaffian matrix are eliminated in every iteration. Section 2 provides an overview of the ABS methods. Then we discuss the new extended reduced rank two update and present the corresponding algorithm for solving compatible systems. We also state and prove some results on the algorithm in this section. In Section 3, we give some specific formulas for updating the Abaffian matrix. In Section 4, we discuss computational complexity of our algorithm.

2. The ABS method and the new extended rank two update

The ABS methods, introduced by Abaffy, Broyden and Spedicato [2], are a general class of algorithms for solving linear and nonlinear algebraic systems and have been extensively applied to several types of linear systems [5,6]. The basic ABS algorithm starts with an initial vector $x_0 \in \mathbb{R}^n$ (arbitrary) and a nonsingular matrix $H_0 \in \mathbb{R}^{n \times n}$ (Spedicato’s parameter). Given that $x_i$ is a solution of the first $i$ equations, the ABS algorithm computes $x_{i+1}$ as the solution of the first $i+1$ equations, if compatible, performing the following steps (see [1,2]).

1. Determine $z_i$ (Broyden’s parameter) so that $z_i^T H_i a_i \neq 0$ and set
   $$ p_i = H_i^T z_i. $$

2. Update the solution by
   $$ x_{i+1} = x_i + \alpha_i p_i, $$
   where the step size $\alpha_i$ is given by
   $$ \alpha_i = \frac{b_j - a_i^T x_i}{a_i^T p_i}. $$

3. Update the Abaffian matrix $H_i$ by
   $$ H_{i+1} = H_i - \frac{H_i w_i z_i^T H_i}{w_i^T H_i a_i}, $$
   where $w_i \in \mathbb{R}^n$ (Abaffy’s parameter) is arbitrary provided that $w_i^T H_i a_i \neq 0$.

It is easily observed that the basic ABS method (the above scheme) satisfies a new equation in every iteration. So, at most $m$ iterations are needed to determine a solution of a compatible system.

We now discuss an approach that we have given in [3], to satisfy two equations at a time. Here, we first motivate the idea and then present a new algorithm in the subsequent section. We consider the system (1.1) and we suppose that $\text{rank}(A) = m$, where $m = 2l$ is even. Let

$$ A = [a_1 \ldots a_m]^T, \quad A^{2i} = [a_1 \ldots a_{2i}]^T, \quad b^{2i} = [b_1 \ldots b_{2i}]^T $$

and

$$ r_j(x) = a_j^T x - b_j, \quad j = 1, 2, \ldots, m, $$

where $a_j^T$ is the $j$th row of $A$. Assume that we are in the $i$th step and $x_i$ satisfies $A^{2i} x = b^{2i}$. We determine $H_i \in \mathbb{R}^{l \times n}$ (where $j_i$ is an appropriate index value), $z_i \in \mathbb{R}^l$ and $\gamma_i \in \mathbb{R}$ so that

$$ x_{i+1} = x_i - \gamma_i H_i^T z_i. $$

(2.2)
be a solution of the first $2i + 2$ equations of the system (1.1). That is,

$$A^{2i+2}x_{i+1} = b^{2i+2}, \quad (2.3)$$

or

$$r_j(x_{i+1}) = 0, \quad j = 1, 2, \ldots, 2i + 2.$$

If we have

$$H_ia_j = 0, \quad j = 1, 2, \ldots, i,$$

then it is easily seen that

$$r_j(x_{i+1}) = 0, \quad j = 1, 2, \ldots, i.$$

For $j = 2i + 1$ and $2i + 2$, we must have

$$\begin{align*}
    a_{2i+1}^T(x_i - \gamma_i H_i^T z_i) - b_{2i+1} &= 0, \\
    a_{2i+2}^T(x_i - \gamma_i H_i^T z_i) - b_{2i+2} &= 0,
\end{align*}$$

or

$$\begin{align*}
    \gamma_i (H_ia_{2i+1})^T z_i &= r_{2i+1}(x_i), \\
    \gamma_i (H_ia_{2i+2})^T z_i &= r_{2i+2}(x_i). \quad (2.4)
\end{align*}$$

Suppose that $r_{2i+1}(x_i) \neq 0$ and $r_{2i+2}(x_i) \neq 0$. Then $\gamma_i$ must be nonzero and (2.4) is compatible if and only if (with appropriate choices of $H_i$ and $z_i$) we have:

$$\frac{r_{2i+1}(x_i)}{(H_ia_{2i+1})^T z_i} = \frac{r_{2i+2}(x_i)}{(H_ia_{2i+2})^T z_i}. \quad (2.5)$$

Note that there are several ways to satisfy the above relation. We consider the following process. First, we scale the equations $2i + 1$ and $2i + 2$ by the factors $r_{2i+2}(x_i)$ and $r_{2i+1}(x_i)$, respectively, and then replace the original equations. Thus, we let

$$\begin{align*}
    a_{2i+1} &= r_{2i+2}(x_i)a_{2i+1}, \\
    a_{2i+2} &= r_{2i+1}(x_i)a_{2i+2}, \quad (2.6)
\end{align*}$$

and

$$\begin{align*}
    b_{2i+1} &= r_{2i+2}(x_i)b_{2i+1}, \\
    b_{2i+2} &= r_{2i+1}(x_i)b_{2i+2}. \quad (2.7)
\end{align*}$$

It is clear that the new residuals are equal; that is,

$$r_{2i+1}(x_i) = r_{2i+2}(x_i) = r_{2i+2}(x_i)r_{2i+1}(x_i). \quad (2.8)$$

Using (2.8), the relation (2.5) is written as

$$\gamma_i = \frac{r_{2i+1}(x_i)}{(H_ia_{2i+1})^T z_i} = \frac{r_{2i+2}(x_i)}{(H_ia_{2i+2})^T z_i}. \quad (2.9)$$

There are several ways to satisfy (2.9); for example,

1. Choose an appropriate update for $H_i$ so that $H_ia_{2i+1} = H_ia_{2i+2} \neq 0$.
2. Choose a vector $z_i$ from the space orthogonal to the vector $H_i(a_{2i+2} - a_{2i+1})$.  

Please cite this article as: K. Amini et al., Extended reduced rank two Abaffian update schemes ..., Applied Mathematics and Computation (2006), doi:10.1016/j.amc.2006.06.105
Here we use the first approach. Thus, the matrix $H_i$ must satisfy the following properties:

\[
\begin{align*}
H_i a_1 &= 0, \\
&\vdots \\
H_i a_{2i} &= 0, \\
H_i a_{2i+1} &= H_i a_{2i+2}.
\end{align*}
\] (2.10)

Now we let

\[
c_i = \begin{cases} 
  a_1, & i = 1, \\
  a_i - a_{i-1}, & i > 1.
\end{cases}
\] (2.11)

Using (2.11), the system (2.10) is written as follows:

\[
\hat{c}_j = H_i c_j = 0, \quad j = 1, 2, \ldots, 2i \quad \text{and} \quad j = 2i + 2.
\] (2.12)

**Remark 1.** Note that $H_i e_{2i+1} = H_i (a_{2i+1} - a_{2}) = H_i a_{2i+1} = H_i a_{2i+2}$.

So, to compute $H_{i+1}$ from $H_i$, it is sufficient that the relations (2.12) hold. Since two new equations are considered in each step, we use an extended rank two update for the Abaffian matrix. We proceed by induction. Suppose that the matrix $H_i$ satisfies (2.12). We need to have

\[
H_{i+1} c_j = 0, \quad j = 1, 2, \ldots, 2i + 2 \quad \text{and} \quad j = 2i + 4.
\] (2.13)

To establish (2.13), we define

\[
H_{i+1} = G_i H_i,
\] (2.14)

and determine $G_i$, so that

\[
\begin{align*}
H_{i+1} c_{2i+1} &= 0, \\
H_{i+1} c_{2i+4} &= 0.
\end{align*}
\]

It is sufficient to let $G_i$ be a $j_{i+1} \times j_i$ matrix satisfying

\[
G_i x = 0 \iff x = \lambda_1 \hat{c}_{2i+1} + \lambda_2 \hat{c}_{2i+4} = \lambda_1 \tilde{a}_{2i+1} + \lambda_2 \tilde{c}_{2i+4}
\] (2.15)

for $\lambda_1$ and $\lambda_2$ arbitrary scalars and $\hat{c}_{2i+1} = H_i e_{2i+1}$, $\hat{c}_{2i+4} = H_i e_{2i+4}$, $\tilde{a}_{2i+1} = H_i a_{2i+1}$. We shall discuss various choices for $G_i$ in Section 3.

To complete the induction, $H_0$ should be chosen so that $H_0 a_1 = H_0 a_2$, or $H_0 c_2 = 0$. Suppose that $\tilde{H}_0$ be an arbitrary nonsingular matrix. We obtain $H_0$ from $\tilde{H}_0$ by using an extended rank one update. We let

\[
H_0 = \tilde{G}_0 \tilde{H}_0,
\] (2.16)

where $\tilde{G}_0$ is a $j_0 \times n$ matrix such that the following property holds:

\[
\tilde{G}_0 x = 0 \iff x = \lambda \tilde{c}_2,
\] (2.17)

where $\lambda$ is an arbitrary scalar and $\tilde{c}_2 = \tilde{H}_0 c_2$. Therefore, we proved the following theorem.

**Theorem 1.** Given $(m = 2l)$ arbitrary linearly independent vectors $a_1, \ldots, a_m \in \mathbb{R}^n$ and an arbitrary nonsingular matrix $\tilde{H}_0 \in \mathbb{R}^{n \times n}$, let $H_0$ be generated by (2.16) with $\tilde{G}_0$ satisfying (2.17) and the sequence of matrices $H_1, \ldots, H_{l-1}$ be generated by (2.14) with $G_i$ satisfying (2.15). Then the following properties hold for $i = 1, \ldots, l - 1$:

(i) $H_i a_j = 0$, \hspace{1cm} $j = 1, \ldots, 2i$,
(ii) $H_i a_{2i+1} = H_i a_{2i+2}$,
(iii) $H_i e_j = 0$, \hspace{1cm} $j = 1, \ldots, 2i$ and $j = 2i + 2$.

Please cite this article as: K. Amini et al., Extended reduced rank two Abaffian update schemes ..., Applied Mathematics and Computation (2006), doi:10.1016/j.amc.2006.06.105
Remark 2. Before we present the algorithm, we need to explain the definition of $\gamma_i$, based on the value of the residuals of the two new equations being considered. We saw that $\gamma_i$ must be nonzero when the corresponding residues are nonzero. We use the following strategy for the definition of $\gamma_i$. If one of the residual values is nonzero and the other is zero, we replace the equations corresponding to the zero residue by the sum of the two equations. Hence, without changing the solution of the original system, the new equation will have the same nonzero residual value as the other equation. But, if both residues are zero, then $\gamma_i$ will be zero and $x_{i+1}$ will be set to $x_i$, as expected.

Now, we can present the steps of the new algorithm for solving full row rank (and hence compatible) systems.

Algorithm 1. Extended ABS method with rank two Abaffian update for solving full row rank linear systems. (Assume that $A_{m \times n}$ is full row rank and $m = 2l$.)

1. (a) Compute $x_1 = r_1(x_0)$ and $\beta_1 = r_2(x_0)$.
2. (a) Let $c_1 = a_1$ and $c_2 = a_2 - a_1$.
3. (b) Compute the $n$-dimensional vector $\bar{c}_2 = \bar{H}_0c_2$ and compute
   \[ H_0 = \bar{G}_0\bar{H}_0, \]
   where $\bar{G}_0$ is a $j_0 \times n$ matrix that satisfies (2.17).
4. (c) Select $z_0 \in \mathbb{R}^{j_0}$ so that $z_0^T H_0 a_1 \neq 0$, and compute
   \[ \gamma_0 = \frac{x_1 b_1}{z_0^T H_0 a_1}, \]
   \[ x_1 = x_0 - \gamma_0 H_0 z_0. \]
5. (b) Compute $c_{i+1} = r_{2i+1}(x_i)$ and $\beta_{i+1} = r_{2i+1}(x_i)$.
6. If $x_{i+1} = 0$ and $\beta_{i+1} = 0$ then let
   \[ a_{2i+1} = a_{2i+1} + a_{2i+2}, \quad b_{2i+1} = b_{2i+1} + b_{2i+2}. \]
   If $x_{i+1} \neq 0$ and $\beta_{i+1} = 0$ then let
   \[ a_{2i+1} = x_{i+1}, \quad a_{2i+2} = a_{2i+1} + a_{2i+2}, \quad b_{2i+2} = b_{2i+1} + b_{2i+2}. \]
   If $x_{i+1} \beta_{i+1} \neq 0$ then let
   \[ a_{2i+1} = \beta_{i+1}a_{2i+1}, \quad a_{2i+2} = x_{i+1} a_{2i+2}, \quad b_{2i+2} = x_{i+1} b_{2i+2}. \]
    (2) Compute
    \[ \gamma_i = \frac{x_i b_i}{z_0^T H_0 a_i}, \]
    \[ x_i = x_{i-1} - \gamma_i H_0 z_0. \]
8. (b) Compute the vector $c_{2i+2} = a_{2i+2} - a_{2i+1}$.
9. Compute the vector $c_{2i+1} = a_{2i+1} + a_{2i+2}$ and $\beta_{2i+1} = r_{2i+1}(x_{i-1})$.
10. Compute the vector $c_{2i+2} = a_{2i+1} + a_{2i+2}$, and compute
    \[ H_i = G_{i-1} H_{i-1}, \]
    where $G_{i-1}$ is a $j_i \times j_{i-1}$ matrix satisfying (2.15).
Remark 3. Note that the matrices $H_i$ are computed by conditions (2.14) and (2.15) for $i = 1, \ldots, l - 1$, and

$$x_i = x_{i-1} - \gamma_{i-1}H_{i-1}^Tz_{i-1}$$

is a solution of the system of equations. To compute the general solution of the system, we need a matrix $H$ with the following properties:

$$Ha_j = 0, \quad j = 1, \ldots, m.$$ 

It can easily be verified that the matrix $H$ can be computed by a final rank one update as below:

$$H = H_l = G_{l-1}H_{l-1},$$

(2.18)

where $G_{l-1}$ is a $j_l \times j_{l-1}$ matrix that satisfies the following property:

$$G_{l-1}x = 0 \iff x = \lambda H_{l-1}a_{2l-1}.$$ 

Thus the general solution of the system is given by

$$x = x_i - H^Ts,$$

where $s \in \mathbb{R}^h$ is arbitrary.

Remark 4. Note that in step (2) of Algorithm 1, an obvious choice for $z_0$ is $H_0a_1$.

Now, we establish some properties of the matrices $H_i$, as generated by Algorithm 1.

**Theorem 2.** Assume that $a_1, \ldots, a_m$ are linearly independent vectors in $\mathbb{R}^n$. Let $\tilde{H}_0 \in \mathbb{R}^{n \times p}$ be an arbitrary nonsingular matrix. $H_0$ be defined as (2.16) and for $i = 1, \ldots, l-1$, the sequence of matrices $H_i$ be generated by (2.14). Then for any $i$, $0 \leq i \leq l - 1$, and $j$, $2i + 2 \leq j \leq m$, the vectors $Ha_j$ are nonzero and linearly independent (or equivalently, $Ha_{2i+1}$ and $Ha_j$, $2i + 3 \leq j \leq m$, are nonzero and linearly independent).

**Proof.** We proceed by induction. For $i = 0$, since the vectors $a_1, \ldots, a_m$ are linearly independent and the matrix $\tilde{H}_0$ is nonsingular, so the vectors $\tilde{H}_0a_1, \ldots, \tilde{H}_0a_m$ are also linearly independent. Now, since the matrix $\tilde{G}_0$ is full row rank and has the property (2.17), so the vectors $\tilde{G}_0\tilde{H}_0a_2, \ldots, \tilde{G}_0\tilde{H}_0a_m$ are nonzero and linearly independent. Therefore, using (2.16), we have the theorem for $i = 0$.

Now we assume that the theorem is true up to $0 \leq k \leq l - 1$, and then prove for $k + 1$. From the induction hypothesis, the vectors $H_ka_j$, for $2k + 2 \leq j \leq m$, are linearly independent. On the other hand the matrix $G_k$, as defined by (2.14), is full row rank and has the property (2.15). So, the vectors $G_kH_ka_{2k+4}, \ldots, G_kH_ka_m$ are nonzero and linearly independent (the statement in the parenthesis in Theorem 2 is now simply verified by the fact that $Ha_{2i+1} = H_ia_{2i+2}$).

**Corollary 1.** If $i < \frac{n}{2}$ and the vectors $a_1, \ldots, a_{2i+2}$ are linearly independent, then $Ha_{2i+1} = H_ia_{2i+2} \neq 0$, and there exists $z_i \in \mathbb{R}^h$ such that $z_i^TH_ia_{2i+2} \neq 0$.

The proof of the following lemma is obvious.

**Lemma 1.** The vectors $a_1, \ldots, a_m$ are linearly independent if and only if the vectors $c_1, \ldots, c_m$ are linearly independent.
Now, using Lemma 1, we have the following result.

**Theorem 3.** For the matrices $H_i$ given by (2.14), (2.16) and (2.18), we have

\[
\dim R(H_i) = n - 2i - 1, \quad 0 \leq i \leq l - 1,
\]
\[
\dim N(H_i) = 2i + 1, \quad 0 \leq i \leq l - 1,
\]
\[
\dim R(H_l) = n - m,
\]
\[
\dim N(H_l) = m.
\]

An interesting question of concern arises when $H_i a_{2i+1} = H_i a_{2i+2} = 0$. Theorem 4 below shows this to be equivalent to the vectors $a_1, \ldots, a_{2i+2}$ being linearly dependent.

**Theorem 4.** Assume $a_1, \ldots, a_{2i}$ are linearly independent. Assume $H_i$ can be defined from $H_{i-1}$ according to (2.14). Then $H_i a_{2i+1} = H_i a_{2i+2} = 0$, if and only if $a_1, \ldots, a_{2i+2}$ are linearly dependent.

**Proof.** By Corollary 1, if $H_i a_{2i+1} = 0$ then the vectors $a_1, \ldots, a_{2i+2}$ are linearly dependent. To prove the converse, for $i = 0$, let $a_2 = xa_1$, $x \neq 1$ (for $x = 1$, it is easily verified that $a_2 = a_1$, which cannot allow the definition of $H_0$). We know,

\[
0 = H_0 a_2 = H_0 (a_2 - a_1) = H_0 (xa_1 - a_1) = (x - 1)H_0 a_1.
\]

This implies that $H_0 a_1 = H_0 a_2$. For $i \geq 1$, since $a_1, \ldots, a_{2i}$ are linearly independent, then the dependence of $a_1, \ldots, a_{2i+2}$ can happen in one of the following ways:

(i) $a_{2i+1}$ or $a_{2i+2}$ is linearly dependent on $a_1, \ldots, a_{2i}$, or
(ii) $a_{2i+1}$ and $a_{2i+2}$ are linearly dependent.

In case (i), let us assume, without loss of generality, that

\[
a_{2i+1} = \sum_{j=1}^{2i} a_j a_j.
\]

Then, using the fact that $H_i a_j = 0$, $j = 1, \ldots, 2i$, we have

\[
H_i a_{2i+1} = H_i \left( \sum_{j=1}^{2i} a_j a_j \right) = \sum_{j=1}^{2i} a_j H_i a_j = 0.
\]

In case (ii), let

\[
a_{2i+2} = xa_{2i+1}, \quad x \neq 1
\]

(for $x = 1$, we have $a_{2i+2} = a_{2i+1}$ which implies that $c_{2i+2} = 0$ and hence $H_i$ cannot be defined from $H_{i-1}$, contradicting the assumption of the theorem). Then, using the fact that $H_i c_{2i+2} = 0$, we have

\[
0 = H_i a_{2i+2} - H_i a_{2i+1} = H_i (xa_{2i+1}) - H_i a_{2i+1} = (x - 1)H_i a_{2i+1},
\]

which shows $H_i a_{2i+1} = 0$. \(\square\)

3. Choices for $G_k$

The size of $G_k$, $j_k$, is a parameter in the new presented class. It seems that one of the most reasonable and one of the best choices is the minimum size, which leads to a subclass defined by $j_k = n - 2k - 1$. We give now some specific construction formulas for $G_k$, as examples, in this subclass.
3.1. The first choice

In [3], we present a class of methods for solving full row rank linear systems using a rank two update of the Abaffian matrix. This method is a subclass of our new method when \( G_k^{(1)} \) is selected to be an \( n \times n \) matrix 

\[
\tilde{G}_0^{(1)} = I - \tilde{H}_0 c_2 \tilde{w}_0^T,
\]

and for \( k = 1, \ldots, l - 1, \)

\[
G_k^{(1)} = I - H_k a_{2k+1} w_k^T - H_k c_{2k+4} \tilde{w}_k^T,
\]

\[
= I - H_k c_{2k+4} \tilde{w}_k^T - H_k a_{2k+1} w_k^T, \tag{3.19}
\]

where \( w_0 \) is an arbitrary vector satisfying \( \tilde{w}_0^T H_0 c_2 = 1, \( w_k \) and \( \tilde{w}_k \) are arbitrary vectors save the conditions

\[
\begin{align*}
& \begin{cases}
 w_k^T H_k a_{2k+1} = 1, \\
 w_k^T H_k c_{2k+4} = 0,
\end{cases} \\
& \begin{cases}
 \tilde{w}_k^T H_k a_{2k+1} = 0, \\
 \tilde{w}_k^T H_k c_{2k+4} = 1.
\end{cases}
\end{align*} \tag{3.20}
\]

We show that the matrix \( G_k^{(1)} \) defined by (3.19) has the property specified by (2.15).  

**Lemma 2.** For an \( n \) dimensional vector \( x \), \( G_k^{(1)} x = 0 \) if and only if

\[
\exists \lambda_1, \lambda_2 : x = \lambda_1 a_{2k+1} + \lambda_2 c_{2k+4}
= \lambda_1 H_k a_{2k+1} + \lambda_2 H_k c_{2k+4}.
\]

**Proof.** For \( x = \lambda_1 a_{2k+1} + \lambda_2 c_{2k+4} \), by (3.20) we have,

\[
G_k^{(1)} x = \lambda_1 (I - H_k a_{2k+1} w_{k}^T - H_k c_{2k+4} \tilde{w}_k^T) H_k a_{2k+1} + \lambda_2 (I - H_k a_{2k+1} w_{k}^T - H_k c_{2k+4} \tilde{w}_k^T) H_k c_{2k+4} = 0.
\]

Conversely, let \( G_k^{(1)} x = 0 \). It implies that

\[
0 = G_k^{(1)} x = (I - H_k a_{2k+1} w_{k}^T + H_k c_{2k+4} \tilde{w}_k^T) x = x - (w_k^T x) a_{2k+1} + (\tilde{w}_k^T x) c_{2k+4},
\]

or

\[
x = \lambda_1 a_{2k+1} + \lambda_2 c_{2k+4},
\]

where \( \lambda_1 = w_k^T x \) and \( \lambda_2 = \tilde{w}_k^T x \) are scalars. \( \square \)

Now, assume that the vectors \( a_1, \ldots, a_m \) are linearly independent. According to Theorem 3, we have \( \dim \mathcal{N}(H_k) = 2k + 1 \) and hence \( 2k + 1 \) rows of the matrix \( H_k \) are dependent on other rows of \( H_k \). Knowing this, we can define \( H_k \) in such a way that exactly \( 2k + 1 \) rows of \( H_k \) are the zero vector. Assume that the rows \( I_1, \ldots, I_{2k+1} \) of \( H_k \) are the zero vector. From the update \( H_k + 1 = G_k^{(1)} H_k \), it is clear that the same rows \( I_1, \ldots, I_{2k+1} \) are also the zero vector. In [3], we have chosen the parameters \( w_k \) and \( \tilde{w}_k \) so that two new rows \( r \) and \( s \) of \( H_{k+1} \) also become the zero vector. So, we can define the \( (n - 2k - 3) \times (n - 2k - 1) \) matrix \( \tilde{G}_k^{(1)}, \)

\[
\tilde{G}_k^{(1)} = P_{rs} (I - a_{2k+1} w_k^T - c_{2k+4} \tilde{w}_k^T), \tag{3.21}
\]

where \( I \) is the \( (n - 2k - 1) \times (n - 2k - 1) \) matrix obtained from the unit matrix by deletion of the rows and columns \( I_1, \ldots, I_{2k+1} \), and \( P_{rs} \) is an operator denoting the deletion of rows \( r \) and \( s \).

3.2. The second choice

Setting \( j_k = n - 2k + 1 \) and making use of two Householder transformations, we define matrix \( G_k^{(2)} \) by

\[
\tilde{G}_0^{(2)} = T_0 \left( I - \frac{2u_0 u_0^T}{\|u_0\|^2} \right),
\]

Please cite this article as: K. Amini et al., Extended reduced rank two Abaffian update schemes ..., Applied Mathematics and Computation (2006), doi:10.1016/j.amc.2006.06.105
where $T_0$ and $u_0$ is defined as
\[ T_0 = (0_{(n-1)\times 2}, I_{n-1}), \quad u_0 = c_2 + (\text{sign}(x_2)\|c_2\|)e_1, \]
and for $k = 1, \ldots, l - 1$,
\[ G_k^{(2)} = T_k \left[ \left( I - \frac{2uu^T}{\|u\|^2} \right) \left( I - \frac{2vv^T}{\|v\|^2} \right) \right], \tag{3.22} \]
where $T_k$ is a $(n-2k-1) \times (n-2k+1)$ matrix of the form
\[ T_k = (0_{(n-2k-1)\times 2}, I_{n-2k-1}), \tag{3.23} \]
and $u$ and $v$ are defined as
\[ u = c_{2k+1} + (\text{sign}(x_{2k+2})\|c_{2k+1}\|)e_1, \]
and
\[ v = c_{2k+4} + (\text{sign}(\beta_{2k+2})\|c_{2k+4}\|)e_2 \]
with $e_1$ and $e_2$ being the first and the second unit vector in $\mathbb{R}^{n-2k-1}$, respectively, and
\[ c_{2k+1} = H_k c_{2k+1} = (x_{2k+2}, \ldots, x_n)^T, \tag{3.24} \]
\[ \bar{c}_{2k+4} = H_k c_{2k+4} = (\beta_{2k+2}, \ldots, \beta_n)^T. \tag{3.25} \]
It is easy to verify that $G_k^{(2)}x = 0$ if and only if $x = \lambda_1 a_{2k+1} + \lambda_2 c_{2k+4}$.

**Remark 5.** If we define $x = 2\|u\|^2$, $\beta = 2\|v\|^2$ and $P = I - xxu^T$, then we have
\[ (I - xxu^T)(I - \beta vv^T) = I - xxu^T - \beta(Pv)v^T = I - [xu \beta Pv][uv]^T. \]
So, $G_k^{(2)}$ can be written as
\[ G_k^{(2)} = T_k(I - [xu \beta Pv][uv]^T). \]

### 3.3. The third choice

Assume that $a_1, \ldots, a_m$ are linearly independent. Let $a_{2k+1} = (x_{2k+2}, \ldots, x_n)^T$ and $c_{2k+4} = (\beta_{2k+2}, \ldots, \beta_n)^T$. We can find indices $r$ and $s$ so that
\[ x_r \beta_s - a_r \beta_s \neq 0, \tag{3.26} \]
because if such indices do not exist, then it is deduced that the vectors $a_{2k+1}$ and $c_{2k+4}$ are linearly dependent. This, however, implies that the vectors $H_k a_{2k+2}$, $H_k a_{2k+3}$ and $H_k a_{2k+4}$ are linearly dependent, the latter contradicting Theorem 2. Now, construct $G_k^{(3)}$ by
\[ \tilde{G}_k^{(2)} = (q_0 \quad I_{n-1}), \]
where $q_0$ is defined as $q_0 = (\tilde{x}_2/\tilde{x}_1, \ldots, \tilde{x}_2/\tilde{x}_1)^T$ where
\[ \tilde{c}_2 = (\tilde{x}_1 \cdots \tilde{x}_n)^T, \]
and for $k = 1, \ldots, l - 1$,
\[ G_k^{(3)} = P_k(T_k \quad I_{n-2k-1}), \tag{3.27} \]
where $P_k$ is a permutation matrix that sets the first and second columns of $T_k$ in column $r$ and $s$ of $G_k^{(3)}$, respectively, and $T_k$ is a $(n - 2k - 1) \times 2$ matrix to be determined with the property,
\[ P_k(T_k \quad I_{n-2k-1})(a_{2k+1} \quad \tilde{c}_{2k+4}) = 0. \tag{3.28} \]
Hence, one choice for the matrix $T_k$ is:

$$T_k = \frac{1}{x_i \beta_i - x_j \beta_j} \begin{pmatrix} \alpha_i \beta_{2k+2} - \beta_i \alpha_{2k+2} & \alpha_{2k+2} \beta_r - \alpha_r \beta_{2k+2} \\ \vdots & \vdots \\ \alpha_i \beta_i - \beta_i \alpha_i & \alpha_i \beta_r - \alpha_r \beta_i \\ \vdots & \vdots \\ \alpha_i \beta_n - \beta_n \alpha_i & \alpha_n \beta_r - \alpha_r \beta_n \end{pmatrix},$$

where

$$i \in \{2k+2, \ldots, n\} \setminus \{r, s\}.$$ 

It is clear that $G_k^3$ satisfies (2.15).

### 4. Computational complexity

In this section, we provide the total number of multiplications required to compute $G_k H_k$, for $k = 0, \ldots, l - 1$, for each of the $G_k^{(i)}$, $1 \leq i \leq 3$, and $G_k^{(1)}$ presented in the previous section. We also compute the total number of multiplications required for Algorithm 1 when $G_k = G_k^{(1)}$.

**Case i:** $G_k = G_k^{(1)}$.

For iteration $k$, $k = 1, \ldots, l - 1$, the number of multiplications required to compute $G_k^{(1)} H_k$, not considering the multiplications for computing $w_k$ and $\tilde{w}_k$, is $4n^2 + 2n$. So, the total number of multiplications required to compute the product $G_k^{(1)} H_k$, for $k = 0, \ldots, l - 1$, is:

$$N = 2n^2 + O(n) + \sum_{k=1}^{l-1} (4n^2 + O(n)) = 2n^2 + 4n^2(l - 1) + O(nl).$$

Since $l = \frac{n}{2}$, then the total number of multiplications for the case $A_{m,n}$ is $2mn^2 + O(n^2) + O(nl)$.

**Case ii:** $G_k = G_k^{(1)}$.

For iteration $k$, $k = 1, \ldots, l - 1$, the number of multiplications required to compute $G_k^{(1)} H_k$, with the special choices for $w_k$ and $\tilde{w}_k$ as in [3], is $4n + 2n(n - 2k - 1)$. So, the total number of multiplications to compute the product $G_k^{(1)} H_k$, for $k = 0, \ldots, l - 1$, is:

$$N = 2n^2 + \sum_{k=1}^{l-1} (4n + 2n(n - 2k - 1)) = 2n^2l - 2nl^2 + O(n^2).$$

Therefore, the total number of multiplications for the case $A_{m,n}$ is:

$$mn^2 - \frac{1}{2} mn^2 + O(n^2).$$

**Case iii:** $G_k = G_k^{(2)}$.

The number of multiplications required to compute $G_k^{(2)} H_k$ in iteration $k$ is $2(n - 2k - 1)(n - k)$. So, the total number of multiplications, for $k = 0, \ldots, l - 1$, is:

$$N = O(n^2) + \sum_{k=1}^{l-1} 2(n - 2k - 1)(n - k).$$

Therefore, the total number of multiplications for the case $A_{m,n}$ is:

$$mn^2 - \frac{3}{4} mn^2 + \frac{1}{6} m^3.$$

**Case iv:** $G_k = G_k^{(3)}$.

The number of multiplications required to compute $G_k^{(3)} H_k$ in iteration $k$ is $2 + 2(n - 2k - 1)(n - 2k + 1)$. So, the total number of multiplications, for $k = 0, \ldots, l - 1$, is:

Please cite this article as: K. Amini et al., Extended reduced rank two Abaffian update schemes ..., Applied Mathematics and Computation (2006), doi:10.1016/j.amc.2006.06.105
\[ N = O(n^3) + \sum_{k=1}^{l-1} (2 + 2(n - 2k - 1)(n - 2k + 1)) = 2n^2l - 4nl^2 + \frac{8}{3}l^3 + O(n^2) + O(m^2) + O(nl). \]

Therefore, the total number of multiplications for the case \( A_{m|cn} \) is:

\[ mn^2 - nm^2 + \frac{1}{3}m^3 + O(n^2) + O(m^2) + O(mn). \]

Now, we compute the total number of multiplications for Algorithm 1 when we use

\[ G_{i-1} = \overline{G}_{i-1}^{(1)} = P_{r,s}(\overline{T} - \bar{a}_{2i-1}w_{i-1}^T + \bar{c}_{2i+1}\bar{w}_{i-1}^T), \]

in step (7) and

\[ \tilde{G}_0 = P_{j_0}(\overline{T} - \bar{c}_2\bar{w}_0^T), \]

in step (2), where \( w_{i-1}, \bar{w}_{i-1} \in \mathbb{R}^{n-2i+1}, \tilde{w}_0 \in \mathbb{R}^n, P_{j_0} \) and \( P_{r,s} \) are defined in [3]. With these choices, we showed in [3] that the total number of multiplications for the case \( A_{m|cn} \) is:

\[ 3mn^2 - \frac{7}{4}m^3 n + \frac{1}{6}m^3 + O(mn) + O(m^2) + O(n^2). \]

We note that the algorithm of Huang, when implemented with care, requires \( \frac{1}{2}mn^2 + O(mn) \) multiplications [2,4]. Comparing this with our result, we see that the above version of Algorithm 1 requires less work than the Huang’s method when \( m \) gets close to \( n \). In fact, for square systems \( (m = n) \), the leading terms for Algorithm 1 amount to \( \frac{11}{24}n^3 \) as opposed to \( \frac{1}{2}n^3 \) for the Huang’s method. Of course, when \( m \) and \( n \) are not too large, the lower order terms of the computation time will also affect the efficiency.

Acknowledgements

The last two authors thanks The Research Council of Sharif University of Technology for its support. The first author thanks The Research Council of Razi University for its support.

References