BULL. AUSTRAL. MATH. SOC. 65F05, 65F30
VOL. 70 (2004) [17–34]

ABS-TYPE METHODS FOR SOLVING FULL ROW RANK LINEAR SYSTEMS USING A NEW RANK TWO UPDATE

K. AMINI, N. MAHDAVI-AMIRI AND M.R. PEYGHAMI

ABS mthods are direct iteration methods for solving linear systems where the i-th iterate satisfies the first i equations, and therefore a system of m equations is solved in at most m ABS steps. In this paper, using a new rank two update of the Abaffian matrix, we introduce a class of ABS-type methods for solving full row rank linear equations, where the i-th iterate solves the first $2i$ equations. So, termination is achieved in at most $\lfloor (m+1)/2 \rfloor$ steps. We also show how to decrease the dimension of the Abaffian matrix by choosing appropriate parameters.

1. INTRODUCTION

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

$$Ax = b, \tag{1.1}$$

where $A = [a_1, \ldots, a_m]^T$, $a_i \in R^n$, $1 \leqslant i \leqslant m$, $x \in R^n$, $b \in R^m$. The basic ABS methods determine the solution of (1.1) or signify lack of its existence in at most m iterations. In any iteration, one extra equation, if compatible, is satisfied.

Here we suggest an approach based on ABS methods, which in any iteration, two new equations (if compatible) are satisfied. This uses a new update for the Abaffian matrix. On m linearly independent equations, this approach provides a class of algorithms that stop after at most $\lfloor (m+1)/2 \rfloor$ iterations. We have implemented a version of the new algorithms along with the Huang, modified Huang and LU decomposition algorithms, and tested the programs on various problems. Numerical results indicate the competitiveness of the proposed algorithm.

Section 2 provides an overview of the ABS methods. There we discuss a new rank two update and present a new algorithm for solving compatible systems. We also prove some

Received 17th September, 2003
The authors thank the Research Council of Sharif University of Technology for its support. The second author also acknowledges the partial support of Joseph Liu and the hospitality of Computer Science and Engineering Department at York University during his sabbatical year at York University.

results about the algorithm in this section. In Section 3, we describe a modified algorithm, choosing certain parameters to compress the Abaffian appropriately. In Section 4, we discuss computational and numerical results. The algoirthm proposed in Section 3 has been implemented and tested on various problems. The results can be compared with our implementations of the Huang, modified Huang, and LU algorithms.

2. The ABS method and the new rank two update

The basic ABS algorithm starts with an intial vector $x_0 \in R^n$ (arbitrary) and a nonsingular matrix $H_0 \in R^{n\times n}$ (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 by the folowing steps (see [**1, 2**] or [**4**]):

(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 α_i is given by

$$\alpha_i = \frac{b_i - a_i^T x_i}{a_i^T p_i}.$$

(3) Update the Abaffian matrix H_i by

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

where $w_i \in R^n$ (Abaffy's parameter) is arbitrary, provided that

$$w_i^T H_i a_i \neq 0.$$

It is easily observed that the ABS methods satify a new equation at each iteration. So, at most m iterations are needed to determine a solution or signify the lack of it.

We now discuss an approach to satisfy two equations at a time (another approach, different in its choice of parameters, can be seen in [**3**]). Here, we first motivate the idea and then present a new algorithm in the subsequent section. We consider the system (1.1) and we assume that $\operatorname{rank}(A) = m$, where $m = 2l$ is even.

REMARK 1. Note that if m is odd, we can consider the augmented system

$$\begin{pmatrix} A & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix},$$

that contains the same solution x as (1.1). Alternatively, one can use a rank one update at the final iteration.

We shall see that if a solution exists, it is found in at most $m/2$ iterations. Let

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

and

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

Assume that we are the i-th step and x_i satisfies $A^{2i}x = b^{2i}$. We determine $H_i \in R^{n\times n}$, $z_i \in R^n$ and $\gamma_i \in R$ so that

$$x_{i+1} = x_i - \gamma_i H_i^T z_i \tag{2.1}$$

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

$$A^{2i+2} x_{i+1} = b^{2i+2}, \tag{2.2}$$

or

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

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

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

or

$$\begin{cases} \gamma_i (H_i a_{2i+1})^T z_i = r_{2i+1}(x_i), \\ \gamma_i (H_i a_{2i+2})^T z_i = r_{2i+2}(x_i). \end{cases} \tag{2.3}$$

Suppose that $r_{2i+1}(x_i) \neq 0$ and $r_{2i+2}(x_i) \neq 0$. Then γ_i must be nonzero and (2.3) is compatible if and only if we have

$$\frac{r_{2i+1}(x_i)}{(H_i a_{2i+1})^T z_i} = \frac{r_{2i+2}(x_i)}{(H_i a_{2i+2})^T z_i}. \tag{2.4}$$

Note that there are several ways to satisfy the above relation. We consider the following. 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 corresponding equations. Thus, we let

$$\begin{cases} a_{2i+1} = r_{2i+2}(x_i) a_{2i+1}, & b_{2i+1} = r_{2i+2}(x_i) b_{2i+1}, \\ a_{2i+2} = r_{2i+1}(x_i) a_{2i+2}, & b_{2i+2} = r_{2i+1}(x_i) b_{2i+2}. \end{cases} \tag{2.5}$$

https://doi.org/10.1017/S0004972700035784 Published online by Cambridge University Press

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

$$\overline{r}_{2i+1}(x_i) = \overline{r}_{2i+2}(x_i) = r_{2i+1}(x_i)r_{2i+2}(x_i). \tag{2.6}$$

Using (2.6), the relation (2.4) is written as:

$$\gamma_i = \frac{\overline{r}_{2i+1}(x_i)}{(H_i a_{2i+1})^T z_i} = \frac{\overline{r}_{2i+2}(x_i)}{(H_i a_{2i+2})^T z_i}. \tag{2.7}$$

There are several ways to satisfy (2.7); for example,

(1) choose an appropriate update for H_i so that $H_i a_{2i+1} = H_i a_{2i+2} \neq 0$, or

(2) choose a vector z_i from the space orthogonal to the vector

$$H_i(a_{2i+2} - a_{2i+1}).$$

Here we use the first approach. Thus the matirx H_i must satisfy the following properties:

$$\begin{cases} H_i a_1 = 0 \\ \quad \vdots \\ H_i a_{2i} = 0, \\ H_i a_{2i+1} = H_i a_{2i+2}. \end{cases} \tag{2.8}$$

Now we let

$$c_i = \begin{cases} a_1, & i = 1, \\ a_i - a_{i-1}, & i > 1. \end{cases} \tag{2.9}$$

Using (2.9), the system (2.8) is written as:

$$H_i c_j = 0, \quad j = 1, \dots, 2i \ \text{ and } \ j = 2i+2. \tag{2.10}$$

So, to compute H_{i+1} from H_i it will be sufficient that the relations (2.10) hold. Since two new equations are considered in each step, we use a rank two update for the Abaffian matrix. We proceed inductively. Suppose that the matrix H_i satisfies (2.10). We define

$$H_{i+1} = H_i + g_i d_i^T + e_i f_i^T,$$

where $g_i, d_i, e_i, f_i \in R^n$. We need to have

$$H_{i+1} c_j = 0, \quad j = 1, \dots, 2i+2 \ \text{ and } \ j = 2i+4,$$

or equivalently

$$(H_i + g_i d_i^T + e_i f_i^T) c_j = 0, \quad j = 1, \dots, 2i+2 \ \text{ and } \ j = 2i+4.$$

So we must define $g_i, d_i, e_i, f_i \in R^n$ in such a way that

$$H_i c_j + (d_i^T c_j) g_i + (f_i^T c_j) e_i = 0, \quad j = 1, \dots, 2i+2 \text{ and } j = 2i+4. \tag{2.11}$$

By defining

$$d_i = H_i^T w_i, \ f_i = H_i^T \overline{w}_i, \tag{2.12}$$

for some $w_i, \overline{w}_i \in R^n$, the conditions (2.11) are satisfied for $j \leqslant 2i$ and $j = 2i+2$, by the induction hypothesis. Letting $j = 2i+1$ and $j = 2i+4$ in (2.11), we get

$$\begin{aligned} (d_i^T c_{2i+1}) g_i + (f_i^T c_{2i+1}) e_i &= -H_i c_{2i+1}, \\ (d_i^T c_{2i+4}) g_i + (f_i^T c_{2i+4}) e_i &= -H_i c_{2i+4}. \end{aligned} \tag{2.13}$$

We consider the choices

$$e_i = -H_i c_{2i+4}, \quad g_i = -H_i c_{2i+1}, \tag{2.14}$$

with

$$\begin{cases} d_i^T c_{2i+1} = 1, \\ d_i^T c_{2i+4} = 0, \end{cases} \quad \begin{cases} f_i^T c_{2i+1} = 0, \\ f_i^T c_{2i+4} = 1, \end{cases} \tag{2.15}$$

which clearly satisfy (2.13). Now, to satisfy (2.15), w_i and $\overline{w}_i$ may be defined as described below by

$$\begin{cases} w_i^T H_i c_{2i+1} = 1, \\ w_i^T H_i c_{2i+4} = 0, \end{cases} \quad \begin{cases} \overline{w}_i^T H_i c_{2i+1} = 0, \\ \overline{w}_i^T H_i c_{2i+4} = 1. \end{cases} \tag{2.16}$$

NOTES.

(1) $H_i c_{2i+1} = H_i(a_{2i+1} - a_{2i}) = H_i a_{2i+1} - H_i a_{2i}$. Since $H_i a_{2i} = 0$, then $H_i c_{2i+1} = H_i a_{2i+1}$. So, in (2.16), $H_i c_{2i+1}$ may be replaced by $H_i a_{2i+1}$.

(2) It is apparent that the system (2.16) has a solution if and only if the vectors $H_i c_{2i+1} = H_i a_{2i+1}$ and $H_i c_{2i+4} = H_i a_{2i+4} - H_i a_{2i+3}$ are linearly independent. By Theorem 2 below, if the a_i are linearly independent then $H_i a_{2i+1}$ and $H_i c_{2i+4}$ will also be linearly independent and (2.16) will have a solution for all i, and hence the H_i and the x_i are well defined for all i.

Therefore, the updating formula for H_i turns out to be

$$\begin{aligned} H_{i+1} &= H_i - H_i c_{2i+1} w_i^T H_i - H_i c_{2i+4} \overline{w}_i^T H_i \\ &= H_i - H_i a_{2i+1} w_i^T H_i - H_i c_{2i+4} \overline{w}_I^Y H_i, \end{aligned} \tag{2.17}$$

where w_i and $\overline{w}_i$ can be any vectors satisfying (2.16). To complete the induction, H_0 should be chosen so that

$$H_0 a_1 = H_0 a_2,$$

https://doi.org/10.1017/S0004972700035784 Published online by Cambridge University Press

or

$$H_0 c_2 = 0. \tag{2.18}$$

Let $\widehat{H}_0$ be an arbitrary nonsingular matrix. We obtain H_0 from $\widehat{H}_0$ by using a rank one update. Let $H_0 = \widehat{H}_0 - uv^T$, where u, $v \in R^n$ are chosen so that (2.18) is satisfied; that is

$$\widehat{H}_0 c_2 - (v^T c_2) u = 0.$$

This equation is satisfied if we set $u = \widehat{H}_0 c_2$ and $v = \widehat{H}_0^T \widehat{w}_0$, for some $\widehat{w}_0 \in R^n$ which in turn satisfies the condition

$$\widehat{w}_0^T \widehat{H}_0 c_2 = 1. \tag{2.19}$$

It is easily seen that (2.19) can be made to hold with a proper choice of $\widehat{w}_0$, whenever a_0 and a_1 are linearly independent. So, we have

$$H_0 = \widehat{H}_0 - \widehat{H}_0 c_2 \widehat{w}_0^T \widehat{H}_0, \tag{2.20}$$

where $\widehat{w}_0$ is an arbitrary vector satisfying (2.19). Therefore, we proved the following theorem.

Theorem 1. *Given $m = 2l$ arbitrary linearly independent vectors $a_1, \ldots, a_m \in R^n$ and an arbitrary nonsingular matrix $\widehat{H}_0 \in R^{n \times n}$, let H_0 be generated by (2.20) and the sequence of matrices $H_1, \ldots, H_{l-1}$ be generated by (2.17) with w_i and $\overline{w}_i$ satisfying (2.16). Then the following properties hold, for $i = 1, \ldots, l-1$:*

(i) $H_i a_j = 0, \quad j = 1, \ldots, 2i.$

(ii) $H_i a_{2i+1} = H_i a_{2i+2}.$

(iii) $H_i c_j = 0, \quad j = 1, \ldots, 2i$ and $j = 2i + 2.$

Remark 2. Before we present the algorithm, we need to explain the definition of γ_i, based on the value of the residuals of the two new equations being considered. We saw that γ_i must be nonzero when the corresponding residuals are nonzero. We use the following strategy for the defintion of γ_i. If one of the residual values is nonzero and the other is zero, we replace the equation corresponding to the zero residual 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 residuals are zero, then γ_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. (Assume that $A_{m \times n}$ has full row rank and $m = 2l$.)

(0) Let $x_0 \in R^n$ be an arbitrary vector and choose $\widehat{H}_0 \in R^{n \times n}$ (an arbitrary nonsingular matrix). Set $i = 1$.

(1) (a) Compute $\alpha_1 = r_1(x_0)$ and $\beta_1 = r_2(x_0)$.

(b) **If** ($\alpha_1 = 0$ and $\beta_1 \neq 0$) **then** let

$$\alpha_1 = \beta_1,\ a_1 = a_1 + a_2,\ b_1 = b_1 + b_2.$$

If ($\alpha_1 \neq 0$ and $\beta_1 = 0$) **then** let

$$\beta_1 = \alpha_1,\ a_2 = a_1 + a_2,\ b_2 = b_1 + b_2.$$

If $\alpha_1\beta_1 \neq 0$ **then** let

$$\begin{cases} a_1 = \beta_1 a_1, \\ b_1 = \beta_1 b_1, \end{cases} \quad \begin{cases} a_2 = \alpha_1 a_2 \\ b_2 = \alpha_1 b_2. \end{cases}$$

(2) (a) Let $c_2 = a_2 - a_1$.

(b) Select $\widehat{w}_0 \in R^n$ so that $\widehat{w}_0^T \widehat{H}_0 c_2 = 1$ and compute

$$H_0 = \widehat{H}_0 - \widehat{H}_0 c_2 \widehat{w}_0^T \widehat{H}_0.$$

(c) Select $z_0 = R^n$ so that $z_0^T H_0 a_1 \neq 0$, and compute

$$\gamma_0 = \frac{\alpha_1\beta_1}{z_0^T H_0 a_1},$$
$$x_1 = x_0 - \gamma_0 H_0^T z_0.$$

(3) **While** ($i < m/2$) **do** (steps (4)–(8))

(4) Compute $\alpha_{i+1} = r_{2i+1}(x_i)$ and $\beta_{i+1} = r_{2i+2}(x_i)$.

(5) **If** ($\alpha_{i+1} = 0$ and $\beta_{i+1} \neq 0$) **then** let

$$\alpha_{i+1} = \beta_{i+1}, \quad a_{2i+1} = a_{2i+1} + a_{2i+2}, \quad b_{2i+1} = b_{2i+1} + b_{2i+2}.$$

If ($\alpha_{i+1} \neq 0$ and $\beta_{i+1} = 0$) **then** let

$$\beta_{i+1} = \alpha_{i+1}, \quad a_{2i+2} = a_{2i+1} + a_{2i+2}, \quad b_{2i+2} = b_{2i+1} + b_{2i+2}.$$

If $\alpha_{i+1}\beta_{i+1} \neq 0$ **then** let

$$\begin{cases} a_{2i+1} = \beta_{i+1} a_{2i+1}, \\ a_{2i+2} = \alpha_{i+1} a_{2i+2}, \end{cases} \quad \begin{cases} b_{2i+1} = \beta_{i+1} b_{2i+1}, \\ b_{2i+2} = \alpha_{i+1} b_{2i+2}. \end{cases}$$

(6) Compute the vector $c_{2i+2} = a_{2i+2} - a_{2i+1}$.

(7) Select $w_{i-1}, \overline{w}_{i-1} \in R^n$ so that

$$\begin{cases} w_{i-1}^T H_{i-1} a_{2i-1} = 1, \\ w_{i-1}^T H_{i-1} c_{2i+2} = 0, \end{cases} \quad \begin{cases} \overline{w}_{i-1}^T H_{i-1} a_{2i-1} = 0, \\ \overline{w}_{i-1}^T H_{i-1} c_{2i+2} = 1. \end{cases}$$

Compute

$$H_i = H_{i-1} - H_{i-1}a_{2i-1}w_{i-1}^T H_{i-1} - H_{i-1}c_{2i+2}\overline{w}_{i-1}^T H_{i-1}.$$

(8) Select $z_i \in R^n$ so that $z_i^T H_i a_{2i+1} \neq 0$ and compute

$$\gamma_i = \frac{\alpha_{i+1}\beta_{i+1}}{z_i^T H_i a_{2i+1}},$$
$$x_{i+1} = x_i - \gamma_i H_i^T z_i.$$

Set $i = i + 1$.

Endwhile.

(9) **Stop** (x_l is a solution).

REMARK 3. Note that the matices H_i are computed by (2.17) for $i = 1, \dots, l-1$, and

$$x_l = x_{l-1} - \gamma_{l-1}H_{l-1}^T z_{l-1}$$

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

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

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

$$H = H_l = H_{l-1} - H_{l-1}a_{2l-1}w_{l-1}^T H_{l-1}, \tag{2.21}$$

where w_{l-1} is an arbitrary vector satisfying $w_{l-1}^T H_{l-1} a_{2l-1} = 1$ (note that, the a_i being linearly independent, $H_{l-1}a_{2l-1}$ is a nonzero vector, and hence (2.21) is well defined with a proper choice of w_{l-1}). Hence the general solution of the system is given by

$$x = x_l - H^T s,$$

where $s \in R^n$ is arbitrary.

REMARK 4. Note that in step (2) of Algorithm 1, the setting of z_0 and $\widehat{w}_0$ as the vectors $H_0 a_1$ and $c_2/(c_2^T \widehat{H}_0 c_2)$, respectively, are proper.

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

THEOREM 2. *Assume that $a_1, \dots, a_m$ are linearly independent vectors in R^n. Let $\widehat{H}_0 \in R^{n\times n}$ be an arbitrary nonsingular matrix, H_0 be defined as in (2.20), and for $i = 1, \dots, l-1$, the sequence of matrices H_i be generated by (2.17). Then for any i, $0 \leqslant i \leqslant l-1$, and j, $2i+2 \leqslant j \leqslant m$, the vectors $H_i a_j$ are nonzero and linearly independent (or equivalently, $H_i a_{2i+1}$ and $H_i a_j$, $2i+3 \leqslant j \leqslant m$, are nonzero and linearly independent).*

https://doi.org/10.1017/S0004972700035784 Published online by Cambridge University Press

PROOF: We proceed by induction. For $i = 0$, the theorem is true, since if $\sum_{j=2}^{m} \alpha_j H_0 a_j = 0$ then

$$\sum_{j=2}^{m} \alpha_j (\widehat{H}_0 - \widehat{H}_0 c_2 \widehat{w}_0^T \widehat{H}_0) a_j = 0,$$

or

$$\sum_{j=2}^{m} \alpha_j \widehat{H}_0 a_j - \Big(\sum_{j=2}^{m} \alpha_j \widehat{w}_0^T \widehat{H}_0 a_j \Big) \widehat{H}_0 c_2 = 0.$$

By taking $\beta_j = \alpha_j \widehat{w}_0^T \widehat{H}_0 a_j$, for $2 \leqslant j \leqslant m$, we have

$$\sum_{j=2}^{m} \alpha_j \widehat{H}_0 a_j - \Big(\sum_{j=2}^{m} \beta_j \Big) (\widehat{H}_0 a_2 - \widehat{H}_0 a_1) = 0,$$

or

$$\beta \widehat{H}_0 a_1 + (\alpha_2 - \beta) \widehat{H}_0 a_2 + \sum_{j=3}^{m} \alpha_j \widehat{H}_0 a_j = 0,$$

where $\beta = \sum_{j=2}^{m} \beta_j$. Now, since $a_1, \dots, a_m$ are linearly independent and $\widehat{H}_0$ is nonsingular, then $\widehat{H}_0 a_j$, up to $1 \leqslant j \leqslant m$, are linearly independent. Hence $\beta = \alpha_2 = \cdots = \alpha_m = 0$. Therefore the vectors $H_0 a_j$, for $2 \leqslant j \leqslant m$, are linearly independent.

Now we assume that the theorem is true up to k, $0 \leqslant k < l - 1$, and then we prove it to be true for $k+1$. From (2.17), we have, for $2k+4 \leqslant j \leqslant m$,

$$H_{k+1} a_j = H_k a_j - (w_k^T H_k a_j) H_k c_{2k+1} - (\overline{w}_k^T H_k a_j) H_k c_{2k+4}. \tag{2.22}$$

We need to show that the relation

$$\sum_{j=2k+4}^{m} \alpha_j H_{k+1} a_j = 0, \tag{2.23}$$

implies that $\alpha_j = 0$, for $2k+4 \leqslant j \leqslant m$. Using (2.22) we can write (2.23) as follows:

$$\sum_{j=2k+4}^{m} \alpha_j H_k a_j - \Big(\sum_{j=2k+4}^{m} \alpha_j w_k^T H_k a_j \Big) H_k c_{2k+1} - \Big(\sum_{j=2k+4}^{m} \alpha_j \overline{w}_k^T H_k a_j \Big) H_k c_{2k+4} = 0.$$

By taking $\beta_1 = \sum_{j=2k+4}^{m} \alpha_j w_k^T H_k a_j$ and $\beta_2 = \sum_{j=2k+4}^{m} \alpha_j \overline{w}_k^T H_k a_j$, we have

$$\sum_{j=2k+4}^{m} \alpha_j H_k a_j - \beta_1 H_k (\alpha_{2k+1} - a_{2k}) - \beta_2 H_k (a_{2k+4} - a_{2k+3}) = 0.$$

Since from Theorem 1 we have $H_k a_{2k} = 0$ and $H_k a_{2k+1} = H_k a_{2k+2}$, then

$$\sum_{j=2k+4}^{m} \alpha_j H_k a_j - \beta_1 H_k a_{2k+2} + \beta_2 H_k a_{2k+3} - \beta_2 H_k a_{2k+4} = 0,$$

or

$$\sum_{j=2k+5}^{m} \alpha_j H_k a_j - \beta_1 H_k a_{2k+2} + \beta_2 H_k a_{2k+3} + \big(a_{2k+4} - \beta_2\big) H_k a_{2k+4} = 0.$$

Since, by the induction hypothesis, the vectors $H_k a_j$, for $2k+2 \leqslant j \leqslant m$, are linearly independent, we have $\beta_1 = \beta_2 = \alpha_{2k+4} = \alpha_{2k+5} = \cdots = \alpha_m = 0$. Hence, the vectors $H_{k+1} a_j$, for $2k+4 \leqslant j \leqslant m$ are linearly independent (the statement in the parenthesis in Theorem 2 is now simply verified by the fact that $H_i a_{2i+1} = H_i a_{2i+2}$). ∎

Corollary 1. *For all i, $i = 0, 1, \dots, (m/2) - 1$, if the vectors $a_1, a_2, \dots, a_{2i+2}$ are linearly independent, then $H_i a_{2i+1} = H_i a_{2i+2} \neq 0$, and there exists $z_i \in R^n$ such that $z_i^T H_i a_{2i+1} \neq 0$.*

From Note (2), the following corollary is now immediately at hand.

Corollary 2. *If $a_1, a_2, \dots, a_m$ are linearly independent, then the system (2.16) has solution for every i, $0 \leqslant i \leqslant l-2$, and both H_{i+1} and x_{i+1} are well defined.*

The proof of the following lemma is obvious.

Lemma 1. *The vectors $a_1, \dots, a_m$ are linearly independent if and only if the vectors $c_1, \dots, c_m$ are linearly inde[pendent.*

We can now easily prove the following Theorem using Lemma 1.

Theorem 3. *For the matrices H_i given by (2.17), (2.20) and (2.21), we have*

$$\begin{aligned} \dim R(H_i) &= n - 2i - 1, \qquad 0 \leqslant i \leqslant l-1, \\ \dim N(H_i) &= 2i + 1, \qquad 0 \leqslant i \leqslant l-1, \\ \dim R(H_l) &= n - m, \\ \dim N(H_l) &= m. \end{aligned}$$

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 $a_1, \dots, a_{2i+2}$ being linearly independent.

Theorem 4. *Assume $a_1, \dots, a_{2i}$ are linearly independent. Assume H_i can be defined from H_{i-1} according to (2.17) (that is (2.16) has a solution for the case of H_{i-1}). Then $H_i a_{2i+1} (= H_i a_{2i+2}) = 0$, if and only if $a_1, \dots, a_{2i+2}$ are linearly dependent.*

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

$$0 = H_0 c_2 = H_0(a_2 - a_1) = H_0(\alpha a_1 - a_1) = (\alpha - 1) H_0 a_1.$$

This implies that $H_0 a_1 = H_0 a_2 = 0$. For $i \geqslant 1$, since $a_1, \dots, a_{2i}$ are linearly independent, then the dependence of $a_1, \dots, a_{2i+2}$ can happen in any one of the following nonexclusive ways:

(i) a_{2i+1} of 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

$$a_{2i+1} = \sum_{j=1}^{2i} \alpha_j a_j.$$

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

$$H_i a_{2i+1} = H_i\Big(\sum_{j=1}^{2i} \alpha_j a_j\Big) = \sum_{j=1}^{2i} \alpha_j H_i a_j = 0.$$

In case (ii), let

$$a_{2i+2} = \alpha a_{2i+1}, \quad \alpha \neq 1.$$

(For $\alpha = 1$, 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

$$0 = H_i a_{2i+2} - H_i a_{2i+1} = H_i(\alpha a_{2i+1}) - H_i a_{2i+1} = (\alpha - 1) H_i a_{2i+1},$$

which shows $H_i a_{2i+2} = 0$. □

REMARK 5. When $H_i a_{2i+1} = 0$, it is clear that neither x_{i+1} nor H_{i+1} can be defined. In this case, one should identify the cause and propose alternative steps to define x_{i+1} and H_{i+1} (of course one can always make use of the regular rank one ABS steps as alternatives). We also note thet H_{i+1} fails to be defined if and only if the system (2.16) lacks a solution, that is the vectors $H_i c_{2i+1} = H_i a_{2i+1}$ and $H_i c_{2i+4} = H_i a_{2i+4} - H_i a_{2i+3}$ are linearly dependent (the case $H_i a_{2i+1} = 0$ is now a special case here). A similar argument, as given in the proof for Theorem 4, shows that this can happen if and only if $a_1, \ldots, a_{2i+2}, a_{2i+3}, a_{2i+4}$ are linearly dependent.

So, we have the following result.

THEOREM 5. *Assume $a_1, \ldots, a_{2i+2}$ are linearly independent. The system* (2.16) *does not possess a solution if and only if the vectors $a_1, \ldots, a_{2i+4}$ are linearly dependent.*

We emphasise again that if a_i, $1 \leqslant i \leqslant m$, are linearly independent then Theorems 4 and 5 will be irrelevant and the H_i and the x_i are well defined.

Next, we discuss how to economise on the space needed for the Abaffian matrix H_i, and show the reduction of computation time in operations involving the Abaffian matrix.

3. Compression of the Abaffian

Assume that the vectors $a_1, a_2, \ldots, a_m$ are linearly independent. According to Theorem 3, we have $\dim N(H_i) = 2i + 1$ and hence $2i + 1$ rows of the matrix H_i are dependent on other rows of H_i. Knowing this, we can define the H_i in such a way that exactly $2i+1$ rows of H_i are zero. Assume that the rows $I_1, \dots, I_{2i+1}$ of H_i are zero. From (2.17), it is clear that the same rows $I_1, \dots, I_{2i+1}$ of H_{i+1} are also zero. Now, we choose the parameters w_i and $\overline{w}_i$ so that two new rows of H_{i+1} will also become zero.

We note that the parameters $w_i, \overline{w}_i \in \mathbb{R}^{n-2i-1}$ satisfy (2.16). Denoting $e_j = H_i c_j$, we then can write (2.16) as follows:

$$\text{(3.1)} \qquad \begin{cases} w_i^T e_{2i+1} = 1, \\ w_i^T e_{2i+4} = 0, \end{cases} \qquad \begin{cases} \overline{w}_i^T e_{2i+1} = 0, \\ \overline{w}_i^T e_{2i+4} = 1. \end{cases}$$

Now, letting

$$u = w_i = (u_1, \dots, u_{n-2i-1})^T \qquad v = \overline{w}_i = (v_1, \dots, v_{n-2i-1})^T,$$
$$\widehat{e} = e_{2i+1} = (\widehat{e}_1, \dots, \widehat{e}_{n-2i-1})^T, \qquad \overline{e} = e_{2i+4} = (\overline{e}_1, \dots, \overline{e}_{n-2i-1})^T,$$

we can write (2.17) as

$$\begin{aligned} H_{i+1} &= H_i - \left(H_i c_{2i+1} w_i^T - H_i c_{2i+4} \overline{w}_i^T\right) H_i \\ &= H_i - D H_i, \end{aligned}$$

where

$$D = H_i c_{2i+1} w_i^T - H_i c_{2i+4} \overline{w}_i^T = e_{2i+1} w_i^T + e_{2i+4} \overline{w}_i^T = (d_{ij}),$$

and

$$\text{(3.2)} \qquad d_{ij} = \widehat{e}_j u_i + \overline{e}_i v_j.$$

So, we would like to have indices r and s such that $r, s \neq I_1, \dots, I_{2i+1}$, $r \neq s$, and

$$\text{(3.3)} \qquad \begin{cases} 1 - d_{rr} = 0, \\ d_{rj} = 0, \qquad \text{for } j \neq r, \end{cases} \qquad \begin{cases} 1 - d_{ss} = 0, \\ d_{sj} = 0, \qquad \text{for } j \neq s. \end{cases}$$

Let $u_j = v_j = 0$ for $j \neq r, s$. Then using (3.1), we have

$$\text{(3.4)} \qquad \begin{cases} u_r \widehat{e}_r + u_s \widehat{e}_s = 1, \\ u_r \overline{e}_r + u_s \overline{e}_s = 0, \end{cases} \qquad \begin{cases} v_r \widehat{e}_r + v_s \widehat{e}_s = 0, \\ v_r \overline{e}_r + v_s \overline{e}_s = 1. \end{cases}$$

On the other hand, using (3.2), the system (3.3) is written as:

$$\text{(3.5)} \qquad \begin{cases} \widehat{e}_r u_r + \overline{e}_r v_r = 1, \\ \widehat{e}_r u_j + \overline{e}_r v_j = 0, \quad \text{for } j \neq r, \end{cases} \qquad \begin{cases} \widehat{e}_s u_j + \overline{e}_s v_j = 0, \quad \text{for } j \neq s, \\ \widehat{e}_s u_s + \overline{e}_s v_s = 1. \end{cases}$$

https://doi.org/10.1017/S0004972700035784 Published online by Cambridge University Press

Now, since $u_j = v_j = 0$, for $j \neq r, s$, (3.5) can be written as:

$$(3.6) \qquad \begin{cases} \widehat{e}_r u_r + \overline{e}_r v_r = 1, \\ \widehat{e}_r u_s + \overline{e}_r v_s = 0, \end{cases} \qquad \begin{cases} \widehat{e}_s u_r + \overline{e}_s v_r = 0, \\ \widehat{e}_s u_s + \overline{e}_s v_s = 1, \end{cases}$$

It can be shown that the systems (3.4) and (3.6) are equivalent. So, we find r and s so that the two systems (3.4) and (3.6) will have a solution. If we have

$$(3.7) \qquad \widehat{e}_r \overline{e}_s - \overline{e}_r \widehat{e}_s \neq 0,$$

then a unique solution exists for (3.4) and (3.6). One choice for r and s is:

$$\left|\widehat{e}_r \overline{e}_s - \overline{e}_r \widehat{e}_s\right| = M = \max\Big\{\left|\widehat{e}_i \overline{e}_j - \overline{e}_i \widehat{e}_j\right| : \quad i, j \in \{1, 2, \ldots, n\} \backslash \{I_1, \ldots, I_{2i+1}\}\Big\}.$$

Therefore, we have

$$(3.8) \qquad \begin{pmatrix} u_r \\ u_s \end{pmatrix} = \frac{1}{M} \begin{pmatrix} \overline{e}_s \\ -\overline{e}_r \end{pmatrix},$$

and

$$(3.9) \qquad \begin{pmatrix} v_r \\ v_s \end{pmatrix} = \frac{1}{M} \begin{pmatrix} -\widehat{e}_s \\ \widehat{e}_r \end{pmatrix}.$$

It is clear that the solutions (3.8) and (3.9) satisfy (3.4) and (3.6). It now remains to show that indices r and s satisfying (3.7) exist. If such indices do not exist, then it is easily deduced that the vectors $e_{2i+1} = H_i c_{2i+1} = H_i a_{2i+1} = H_i a_{2i+2}$ and $e_{2i+4} = H_i c_{2i+4} = H_i a_{2i+4} - H_i a_{2i+3}$ are linearly dependent. This, however, implies that the vectors $H_i a_{2i+2}$, $H_i a_{2i+3}$ and $H_i a_{2i+4}$ are linearly dependent, contradicting Theorem 2.

The definitions of w_i and $\overline{w}_i$ make two new rows of H_{i+1} the zero vector. Hence, while updating H_i, we can omit these rows. So, we find r and s as above and set $H_{i+1} = P_{r,s}(H_{i+1})$, where $P_{r,s}$ is an operator denoting the deletion of rows r and s.

On the other hand, since $\dim(H_0) = n-1$, we can delete the zero row of H_0 as follows. Since $\widehat{H}_0$ is nonsingular and a_1, a_2 are linearly independent, then $c_2 = a_2 - a_1 \neq 0$ and hence $\widetilde{e} = \widehat{H}_0 c_2 \neq 0$. So, an index j_0 exists such that $\widetilde{e}_{j_0} \neq 0$. We define the vector $\widehat{w}_0 \in R^n$ as:

$$\left|\widetilde{e}_{j_0}\right| = \max_{1 \leqslant i \leqslant n} \left|\widetilde{e}_i\right|,$$

$$(3.10) \qquad \left(\widehat{w}_0\right)_i = \begin{cases} \dfrac{1}{\widetilde{e}_{j_0}}, & \text{if } i = j_0, \\ 0, & \text{if } i \neq j_0. \end{cases}$$

Now, this definition of $\widehat{w}_0$ satisfies (2.19) and makes the row j_0 of H_0 in (2.20) the zero vector. So, we can delete this row from H_0 by an operator P_{j_0}.

From the above discussion we provide the following modified algorithm.

ALGORITHM 2. (Assume that $A_{m\times n}$ has full row rank and $m = 2l$.)

(0) Let $x_0 \in R^n$ be arbitrary and choose $\widehat{H}_0 \in R^{n\times n}$ (an arbitrary nonsingular matrix). Set $i = 1$.

(1) (a) Compute $\alpha_1 = r_1(x_0)$ and $\beta_1 = r_2(x_0)$.

(b) **If** ($\alpha_1 = 0$ and $\beta_1 \neq 0$) **then** let

$$\alpha_1 = \beta_1,\ a_1 = a_1 + a_2,\ b_1 = b_1 + b_2.$$

If ($\alpha_1 \neq 0$ and $\beta_1 = 0$) **then** let

$$\beta_1 = \alpha_1,\ a_2 = a_1 + a_2,\ b_2 = b_1 + b_2.$$

If $\alpha_1\beta_1 \neq 0$ **then** let

$$\begin{cases} a_1 = \beta_1 a_1, \\ b_1 = \beta_1 b_1, \end{cases} \qquad \begin{cases} a_2 = \alpha_1 a_2, \\ b_2 = \alpha_1 b_2. \end{cases}$$

(2) (a) Let $c_2 = a_2 - a_1$.

(b) Compute $\widetilde{e} = \widehat{H}_0 c_2$ and choose the index j_0 so that

$$|\widehat{e}_{j_0}| \max_{1\leqslant i\leqslant n} |\widetilde{e}_i|.$$

(c) Define $\widehat{w}_0 \in R^n$ as (3.10).

(d) Compute $d = \widehat{H}_0 c_2$, and

$$H_0 = P_{j_0}(\widehat{H}_0 - d\widehat{w}_0^T \widehat{H}_0).$$

(P_{j_0} is an operator deleting the j_0-th row of a matrix.)

(e) Select $z_0 \in R^{n-1}$ so that $z_0^T H_0 a_1 \neq 0$, and compute

$$\gamma_0 = \frac{\alpha_1\beta_1}{z_0^T H_0 a_1},$$
$$x_1 = x_0 - \gamma_0 H_0^T z_0.$$

(3) **While** ($i < m/2$) **do** (steps (4)–(9))

(4) Let $\alpha_{i+1} = r_{2i+1}(x_i)$ and $\beta_{i+1} = r_{2i+1}(x_i)$.

(5) **If** ($\alpha_{i+1} = 0$ and $\beta_{i+1} \neq 0$) **then** let

$$\alpha_{i+1} = \beta_{i+1}, \quad a_{2i+1} = a_{2i+1} + a_{2i+2}, \quad b_{2i+1} = b_{2i+1} + b_{2i+2}.$$

If ($\alpha_{i+1} \neq 0$ and $\beta_{i+1} = 0$) **then** let

$$\beta_{i+1} = \alpha_{i+1}, \quad a_{2i+2} = a_{2i+1} + a_{2i+2}, \quad b_{2i+2} = b_{2i+1} + b_{2i+2}.$$

https://doi.org/10.1017/S0004972700035784 Published online by Cambridge University Press

If $\alpha_{i+1}\beta_{i+1} \neq 0$ **then** let

$$\begin{cases} a_{2i+1} = \beta_{i+1}a_{2i+1}, \\ a_{2i+2} = \alpha_{i+1}a_{2i+2}, \end{cases} \qquad \begin{cases} b_{2i+1} = \beta_{i+1}b_{2i+1}, \\ b_{2i+2} = \alpha_{i+1}b_{2i+2}. \end{cases}$$

(6) Compute $c_{2i+2} = a_{2i+2} - a_{2i+1}$, and

$$\begin{cases} \widehat{e} = H_{i-1}a_{2i-1}, \\ \overline{e} = H_{i-1}c_{2i+2}. \end{cases}$$

(7) (a) Choose r and s so that

$$M = \max\Big\{\big|\widehat{e}_i\overline{e}_j - \overline{e}_i\widehat{e}_j\big| : i,\ j \in \{1,\dots,n\}\backslash\{I_1,\dots,I_{2i-1}\}\Big\} = \big|\widehat{e}_r\overline{e}_s - \overline{e}_r\widehat{e}_s\big|.$$

(b) Define w_{i-1} and $\overline{w}_{i-1} \in R^{n-2i+1}$ as below:

$$w_{i-1} = (u_1,\dots,u_{n-2i+1})^T, \quad \overline{w}_{i-1} = (v_1,\dots,v_{n-2i+1})^T,$$

where,

$$u_i = \begin{cases} \dfrac{\overline{e}_s}{M}, & i = r, \\ -\dfrac{\overline{e}_r}{M}, & i = s, \\ 0, & i \neq r, s, \end{cases} \qquad v_i = \begin{cases} -\dfrac{\widehat{e}_s}{M} & i = r, \\ \dfrac{\widehat{e}_r}{M}, & i = s, \\ 0, & i \neq r, s. \end{cases}$$

(8) Compute

$$H_i = P_{r,s}\big(H_{i-1} - H_{i-1}a_{2i-1}w_{i-1}^T H_{i-1} - H_{i-1}c_{2i+2}\overline{w}_{i-1}^T H_{i-1}\big).$$

($P_{r,s}$ is an operator deleting rows r and s of a matrix.)

(9) Select $z_i \in R^{n-2i-1}$ so that $z_i^T H_i a_{2i+1} \neq 0$ and compute

$$\gamma_i = \frac{\alpha_{i+1}\beta_{i+1}}{z_i^T H_i a_{2i+1}},$$
$$x_{i+1} = x_i - \gamma_i H_i^T z_i$$

Set $i = i + 1$.

Endwhile.

(10) **Stop** (x_l is a solution).

REMARK 6. According to step (7), we have

$$w_{i-1}^T H_{i-1} = \frac{\overline{e}_s}{M}h_r^{(i-1)} - \frac{\overline{e}_r}{M}h_s^{(i-1)} = u_r h_r^{(i-1)} + u_s h_s^{(i-1)},$$
$$\overline{w}_{i-1}^T H_{i-1} = \frac{\widehat{e}_r}{M}h_s^{(i-1)} - \frac{\widehat{e}_s}{M}h_r^{(i-1)} = v_s h_s^{(i-1)} + v_r h_r^{(i-1)},$$

https://doi.org/10.1017/S0004972700035784 Published online by Cambridge University Press

Step	Number of multiplications
4	$2n$
5	$2n+2$
6	$2n(n-2i+1)$
7	$(n-2i+1)(n-2i)$
8	$4n+2n(n-2i-1)$
9	$1+n+n(n-2i-1)$

Table 1: Numbers of multiplications required for steps (4)–(9) of Algorithm 2.

where $h_j^{(i-1)}$ is the j-th row of H_{i-1}. Hence we can compute the matrix H_i from H_{i-1}, as follows:

$$H_i = H_{i-1} - \widehat{e}\big[u_r h_r^{(i-1)} + u_s h_s^{(i-1)}\big] - \overline{e}\big[v_s h_s^{(i-1)} + v_r h_r^{(i-1)}\big].$$

REMARK 7. To reduce the computation time in step (9), one can avoid the division by choosing $z_i \in R^{n-2i-1}$ so that $z_i^T H_i a_{2i+1} = 1$.

4. Computational and numerical results

Assume that $A_{m\times n} (m = 2l)$ is a full row rank matrix. We can compute the number of multiplications as follows. The major work is performed in steps (4)–(9) in each iteration. Notice that we need $O(n^2)$ multiplications only for the steps (1) and (2) of Algorithm 2. For iteration i, the number of multiplications required for steps (4)–(9) are summarised in Table 1 (the number of multiplications for steps (8) and (9) were considered using Remarks 6 and 7).

Hence, the total number of multiplications for the l iteration is:

$$\begin{aligned} N &= \sum_{i=1}^{l-1}\big(3n(n-2i-1) + (n-2i+1)(3n-2i) + 9n + 3\big) + O(n^2) \\ &= \sum_{i=1}^{l-1}(6n^2 - 14in + 4i^2) + O(n^2) + O(l^2) + O(nl) \\ &= 6n^2(l-1) - 7nl(l-1) + \frac{2}{3}(l-1)l(2l-1) + O(nl) + O(l^2) + O(n^2). \end{aligned}$$

Since $l = m/2$, then the total number of multiplications for the case $A_{m\times n}$ is:

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

We note that the algorithm of Huang, when implemented with care, requires $(3/2)mn^2 + O(mn)$ multiplications. Comparing this with our result we see that the new class of algorithms 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 2 amount to $(17/12)n^3$ as opposed to $(3/2)n^3$ 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.

We should point out that step 7(a) of Algorithm 2 can be alternatively performed more efficiently. The indices r and s need not to be chosen as the maximal row indices specified by M; instead, we may choose r and s as row indices of two independent rows in $(\widehat{e}\,\overline{e})$. This can easily be done by a number of multiplications at most equal to

$$\sum_{i=1}^{l-1}(n-2i) = O(nm) + O(m^2),$$

and the total cost is hence reduced to:

$$\frac{5}{2}mn^2 - \frac{5}{4}m^2n + O(nm) + O(m^2) + O(n^2).$$

In this case, a comparison of the leading terms of Algorithm 2 and the Huang's algorithm shows that Algorithm 2 needs less work when $m > (4/5)n$ (for $m = n$, the leading term for Algorithm 2 turns out to be $(15/12)n^3$ in contrast to $(3/2)n^3$ for the Huang's method).

Algorithm 2 documented in the previous section along with the Huang, modified Huang, implicit LX and LU algorithms were coded in FORTRAN and tested on various problems. The preliminary numerical results show the efficiency and reliability of Algorithm 2 when m gets close to n. In solving n by n nonsingular systems, we have seen that, while the computation time required for Algorithm 2 is less than the ones for the Huang [**2**, **8**], modified Huang [**2**, **5**, **6**], implicit LX ([**2**, **7**]) and LU algorithms, Algorithm 2 also gives generally more accurate solutions, especially on ill-conditioned problems.

References

[1] J. Abaffy, C.G. Broyden and E. Spedicato, 'A class of direct methods for linear systems', *Numer. Math.* **45** (1984), 361–376.

[2] J. Abaffy and E. Spedicato, *ABS projection algorithms*, Mathematical techniques for linear and nonlinear equations (Ellis Horwood, Chichester, 1989).

[3] M. Adib, N. Mahdavi-Amiri and E. Spedicato, *ABS type methods for solving m linear equations in $m/2$ steps*, Report DMSIA 8/2000 (University of Bergamo, Bergamo, Italy, 2000), pp. 12.

[4] K. Amini and N. Mahdavi-Amiri, 'Solving rank one perturbed linear Diophantine systems by the ABS methods', *Optim. Methods Softw.* (submitted).

[5] E. Spedicato, E. Bodon, A. Del Popolo and N. Mahdavi-Amiri, 'ABS methods and ABSPACK for linear systems and optimization: A review', *4OR* **1** (2003), 51-66.

[6] E. Spedicato, E. Bodon, A. Del Popolo and Z. Xia, 'ABS algorithms for linear systems and optimization: A review and a bibliography', *Ricerca Operativa* **29** (2000), 39-88.

[7] E. Spedicato, Z. Xia and L. Zhang, 'The implicit LX method of the ABS class', *Optim. Methods Softw.* **8** (1997), 99–110.

[8] E. Spedicato and M. Zhu, 'A generalization of the implicit LU algorithm to an arbitrary initial matrix', *Numer. Algorithms* **20** (1999), 343–351.

Department of Mathematical Sciences
Sharif Univeristy of Technology
Tehran
Iran
e-mail: Amini@mehr.sharif.edu
Nezamm@sina.sharif.edu
Peyghami@mehr.sharif.edu